Going Places With The New Apple IIc

All You'll Need to Know to Get There

by Danny Goodman

$3.95 · 53188 · 3 · S & S Electronic Publishing/Pocket Books

0-671-53188-3

Going Places With The New Apple IIc

OTHER BOOKS BY DANNY GOODMAN:

WORD PROCESSING ON THE IBM PC
THE SIMON & SCHUSTER GUIDE TO THE TRS-80 MODEL 100
THE SIMON & SCHUSTER GUIDE TO THE IBM PCjr
THE SIMON & SCHUSTER GUIDE TO THE MACINTOSH (forthcoming)

OTHER BOOKS CO-AUTHORED BY DANNY GOODMAN:

PARENTS GUIDE TO PERSONAL COMPUTERS AND SOFTWARE, with the Editors of Consumer Guide
HOW TO BUY AN IBM PC OR COMPATIBLE COMPUTER, with the Editors of PC World

Going Places With The New Apple IIc

All You'll Need to Know to Get There

by Danny Goodman

PUBLISHED BY
SIMON & SCHUSTER ELECTRONIC PUBLISHING DIVISION
AND POCKET BOOKS
NEW YORK

ALL PHOTOGRAPHS COURTESY OF APPLE COMPUTER, INC.
(see attached list of trademarks)

Published by The Electronic Publishing Division
of Simon & Schuster, Inc. and Pocket Books,
a division of Simon & Schuster, Inc.

Simon & Schuster Building
Rockefeller Center
1230 Avenue of the Americas
New York, New York 10020

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN: 0-671-53188-3

TRADEMARKS

ADAM is a trademark of Coleco Industries, Inc.

Apple, Apple II, Apple II Plus, Apple *//e*, Apple *///*, the Apple logo, Lisa, Macintosh, Mouse Text, Mouse-Paint, MacPaint, AppleSoft, and Imagewriter are trademarks of Apple Computer, Inc.

Bank Street Writer is a trademark of Broderbund Software, Inc.

CompuServe is a registered trademark of CompuServe, Inc., an H&R Block company.

CP/M is a registered trademark of Digital Research Corporation.

Datasoft is a registered trademark of Datasoft, Inc.

dBase II is a trademark of Ashton-Tate.

DIALOG is a service mark of DIALOG Information Services, Inc.

Dow Jones Investor's Workshop and Dow Jones News/Retrieval are trademarks of Dow Jones & Company, Inc.

Electronic Arts and Financial Cookbook are trademarks of Electronic Arts.

Grandma's House is a trademark of Spinnaker Software.

IBM is a registered trademark of International Business Machines Corporation.

Mastering the SAT is a trademark of CBS Inc.

MasterType is a trademark of Lightning Software.

Multiplan and Microsoft are registered trademarks of Microsoft Corporation.

PFS is a registered trademark of Software Publishing Corporation.

Rocky's Boots and The Learning Company are trademarks of The Learning Company.

Stickybear is a trademark of Optimum Resource, Inc.

The Source is a service mark of Source Telecomputing Corp.

ThinkTank is a trademark of Living Videotext, Inc.

Typing Tutor III, Letter Invaders, and Kriya Systems are trademarks of Kriya Systems, Inc.

VisiCalc is a registered trademark of VisiCorp.

WordStar is a registered trademark of MicroPro International Corporation.

Zaxxon is a trademark of Sega Enterprises, Inc.

Zork is a registered trademark of Infocom, Inc.

Acknowledgments

This book would not have been possible without the excellent cooperation received from many individuals at Apple Computer, each of whom played an important role in the writing of the text. I want to thank Ed Colby, Lee Collings, Rob Gemmell, John MacDougall, Kris Olsen, Peggy Redpath, Patty Rodrique, Daniela Salvadori, Doug Solomon, Martha Steffen, Christy Sullivan, and Jim Watts. My deepest appreciation, however, goes to Martha Sessums, who moved mountains on my behalf and made the corporate "operating system" completely transparent to this user. I also extend special thanks to Robert Eckhardt and Frank Schwartz of Simon & Schuster for their confidence in me, and to the dozens of S&S folks who performed miracles to produce the book in a timely manner without sacrificing an ounce of quality.

To Linda

CONTENTS

PART II

What an Apple *II*c Can Do

FOREWORD

by
Stephen G. Wozniak

From the first, the Apple *II* family of computers have been like good friends to their owners. They've performed beautifully as productivity tools, enchanted all of us with challenging, beautiful and exciting games and learning software, and satisfied even the most advanced of us who insisted upon pacing the machines to their limits. As you learned—and we learned—about what a personal computer should be, the machine itself kept growing and improving.

The Apple *II*c is the newest member of the Apple *II* family, and in many ways is the realization of our dream to create a personal computer that can be used and enjoyed by *everyone*. The *II*c will earn its place as the leader in personal computers, and the most important reason for this success will be you: over the years we have learned what Apple *II* users want and expect, and we created the Apple *II*c in response to that feedback. For that reason, the *II*c inherits the rich legacy of Apple *II* hardware and runs thousands of Apple *II* software packages. But you'll also find many exciting new features on the *II*c, incorporating the imagination and creativity of the best minds at Apple.

The Apple *II*c makes personal computing easy, accessible and rewarding. It is light—only 7½ pounds—so you can take it with you anywhere, whenever you want. When you're done working with it at the office,

bring it home and play with your friends and family. It is also very attractive, very powerful and very affordable. You'll be able to set it up in minutes, learn to use it in a few hours, and use it for years. We hope you'll be proud to own it.

The world has not been quite the same since the Apple *//* first appeared—only six years ago. At times I think the story of Apple's success—truly the ultimate American business fantasy—overshadows the way that the computers themselves have touched so many lives, letting people work easier than before, releasing untapped creativity in others, just plain giving enjoyment to still others. When all is said and done, this is the source of my deepest satisfaction—the freedom that Apple's computers have given to people.

I guess I've made my point: using a personal computer can be fun and fascinating. If you already own one, you probably know this. If you are thinking of buying an Apple *//*c, you're in for a real treat.

From our family to yours,

Woz

STEPHEN G. WOZNIAK
Co-Founder of Apple Computers, Inc.
and Inventor of the original Apple *//* computer

INTRODUCTION: For Newcomers

There's a good chance that this is the first book about a personal computer you've ever opened. Only 10 percent of Americans have ever come into direct contact with a personal computer. That number is growing rapidly, however, and computers like the Apple *II*c are contributing to the widespread use of personal computers in the workplace, at school, and especially at home. If you are brand-new to computers, then this introduction is for you. You'll learn what a personal computer can and cannot do, what you need to know to use one productively, and how easily a computer like the Apple *II*c can help you do what you need, or want, to do.

A COMPUTER IS A TOOL

To many people who have never been introduced to them, personal computers are perceived as magical, mystical machines that do the amazing things portrayed on television. J. R. Ewing looks at colorful charts of his oil holdings on his Dallas office computer video screen. "Whiz kids" do mischievous deeds with their computers over telephone hookups. Then come the commercials in between, which depict video-game champs losing out at job interviews because they didn't use a *real* computer, and businesses about to be swept away by competition being saved by the arrival of a personal computer.

Computers don't have to be mysterious and terrifying, though. They are frightening only if you focus your attention on the computer, and not on what it can do for you. Remember when you first learned to ride a bicycle? You were so nervous about trying to coordinate the pedaling, balancing, and steering—and always thinking about how much it would hurt if you fell—that your form was anything but graceful. But gradually, with experience, you overcame those fears until they were completely masked by what the bicycle could do for you: get you from Point A to Point B a lot faster than walking. Applying the bicycle to your transportation needs soon became second nature.

There is a striking parallel between learning to ride a bike and learning to use a computer. When you first unpack the computer from its box, the computer machine and new things like computer disks or printers grab your attention. Gradually, you learn what all the buttons and switches are for, how to hook up the accessories, where the ON switch is, and so on. Eventually, you spend less time worrying about the computer itself, and more time attending to the information that appears on the screen. In other words, you start using the computer as it was intended: a device to help you get from Point A (problem) to Point B (solution) faster or more accurately than you could without it. The computer actually becomes a tool—an extension of your mind and body.

FIVE EASY PIECES

A personal computer is "personal" because it is a computer completely under the control of a single individual. Before the era of personal computers, only

trained data-processing specialists actually operated computers. The kinds of machines they used are called mainframe computers and minicomputers. Mainframes are the giant room-sized monsters that generate the billing statements from your credit card purchases, for example. Minicomputers are typically no larger than an office desk, and keep track of the figures for many medium-sized businesses. As engineers applied new technologies to make computers even smaller and cheaper, they dubbed the desktop boxes *microcomputers* ("micro" is smaller than "mini")—another name for personal computers.

One of the earliest commercially successful personal computers was the original Apple *II*, first available in 1977. Much has happened not only to the Apple *II*, but to personal computers in general since then. Yet the basic elements of a functional personal computer system remain the same, even with such advanced computers as the IBM Personal Computer, Apple Macintosh, and *II*c. You can count the basic elements on the fingers of one hand:

1. central processing unit
2. keyboard
3. display
4. mass-storage system
5. programs that turn the first four elements into a useful tool

The CPU

The core of any computer bears the complex-sounding name *central processing unit*, or CPU for short. Technically speaking, CPU more accurately describes just one of the many integrated circuits (chips) inside the

computer that really runs the whole show. We'll be talking more about the CPU chip inside the *II*c in Chapter 2. But for now, all you need to keep in mind is that all the computing that a computer performs takes place inside the console.

Truly, if any parts of a computer should be mysterious, they would be the circuits inside the computer. They operate at such fast speeds—sending signals around literally millions of times each second—that it is nearly impossible for us to create a mental image of what's going on there. Fortunately, you don't have to know how the computer's circuitry works to use a computer effectively any more than you need to know how fuel injection works to drive an automobile well. All the computer's circuits are "under the hood."

The Keyboard

As you'll see later on, a human and a computer interact with each other. There is a semblance of communication between them, albeit in a rudimentary form. Until such time as low-cost speech-recognition technology allows us to speak in everyday language to a computer, we're limited to a more physical transfer. The most often used device to communicate from human to computer is a keyboard resembling a typewriter keyboard.

Computer keyboards come in two basic styles: integrated and detachable. An integrated keyboard is like the one on the *II*c, in which the keyboard is built into the CPU console, making it easy to transport the computer as a single unit. A detachable keyboard, like those on the IBM PCjr and Apple Macintosh computers, lets you orient the keyboard for the most com-

fortable position, without moving a bulky computer. Some detachable-keyboard users even place the keyboard in their laps for really laid-back typing. When a keyboard is integrated into a lightweight, compact computer like the *II*c, it is not unrealistic to expect users to place the entire computer in their laps. We'll have much more to say about the *II*c keyboard in Chapter 1.

The Display

Just as we need to "talk" to the computer, it must likewise respond to us. Computers are not yet talking freely with the aid of electronic speech synthesis, so they rely on a quieter medium: a display device. Until recently, computer displays have been almost entirely televisionlike video display monitors. Video displays can be anything from an actual home color television set to a special computer monitor to a built-in high-resolution display like that found inside the Macintosh console. Some computer monitors are designed to display color images, while others are better suited for the display of text (letters, numbers, punctuation, etc.).

Video displays, however, are bulky if you need to move the computer around. New flat-panel display technologies are surfacing as alternatives to the video monitor, providing portability and convenience never before possible. Large panel liquid crystal displays (LCDs) can now show as much text as a video monitor. As we'll see in Chapter 3, such a display option is available for the *II*c.

Mass Storage

The fourth piece of our five-part computer system is

called a mass-storage device. For most of what you'll be doing with the computer, you will want to have a record of your work session, because, as we'll see in more detail in Chapter 2, when you turn off the computer, it doesn't remember whatever it was you typed into it earlier. Mass storage keeps a record of your work by saving information in large chunks, rapidly transferring information from the computer to the storage medium and back again when needed. In the //c and most other personal computers today, this device takes the form of a disk drive. The disk drive and the flexible disk that slips inside it let you safely file away information you type into the computer from the keyboard (input) as well as the results of work done by the computer (output) which are displayed on its monitor. The flexible disk medium (also called floppy disk) is much faster and more reliable than the cassette tape used on many inexpensive home computers.

Like monitors and keyboards, disk drives can be either integrated into the console (as in the //c) or housed in a separate cabinet. For portability, having the disk drive integrated into the cabinet means one less piece to have to pack up. Quite often in professional and business applications, a second disk drive is desirable. Some computers have room inside the main console for two disk drives, while other models have you add a second disk drive outside the main unit.

The four pieces of the system we've covered so far—the CPU, keyboard, monitor, and mass-storage device—plus any number of other accessory devices (called peripherals) which plug into the console are lumped together under the term *hardware*. Hardware

consists of the pieces of equipment that perform the same job every time they're called upon. A CPU computes, a disk drive stores information on the disk, and so on. The term hardware is now frequently used in other home electronics categories. In videocassette recorders (VCRs), the actual machine is called the hardware. But what brings a VCR to life is a videotape—called *software*. The same is true for computers. And so, program disks—the last element of a computer system—are called software.

Programs

In its most elemental sense, a computer program is like a script that the computer follows to take on the character intended by the programmer. Just as a good actor has the potential to play virtually any role within certain physical constraints, even if it means adding makeup and wardrobe, a CPU can do almost anything under the direction of the program, although it may require a special kind of printer or other accessory to work properly. In one computing session, the CPU can perform as an intelligent typewriter. Later, the same CPU with a different program becomes a two-way telephone communications terminal. A well-written program should make the CPU's performance convincing enough that you forget you're using a personal computer, but rather an intelligent typewriter or communications terminal.

Today, most programs come stored on floppy disks. To start up a program, you slide the disk into the disk drive, close the disk drive door, and turn on the computer (sometimes you also have to press a few keys to start the program). Your general-purpose computer is instantly turned into a special-purpose com-

puter to help you manage numbers, files, pictures, or words. Without software, the hardware is little more than an expensive paperweight.

A TOOL WITH A PURPOSE

What prevents many people from buying a personal computer is that they can't imagine what they would do with it once they have it. Others may feel pressured to bring a personal computer into their lives because: *(a)* their friends are doing it; *(b)* the kids say they need one; *(c)* they're afraid they'll "fall behind" today's computerized society; *(d)* their business competitors are already computerized; or *(e)* all of the above. Buying a personal computer for the wrong reason is just as dangerous as buying a computer for no reason at all. Both are likely to end up as unsatisfying experiences.

The key to a successful marriage with a personal computer is identifying a legitimate need for the machine before you buy it. It helps to know what a computer is capable of in the first place (in Part II, I'll show you many practical things an Apple *//c* in particular can do). And there are probably telltale signs already, in your workplace and at home, that a personal computer may be useful to you.

Education

One important reason to bring a computer into your home is for educational purposes. Many elementary- and high-school students are introduced to computers at school where they are using the machines both as supplements to the teacher's curriculum and as tools to learn computer programming in languages like

Logo, BASIC, and Pascal. Much of the software used in the schools is also available to consumers for use on the same computers at home. If the child is self-motivated or encouraged to use a computer at home for educational purposes, the computer might provide needed extra help in troublesome subjects, or offer outlets for an otherwise untapped creative talent.

It makes sense for the interested computer student to work on the same computer at home as at school, especially if he or she is studying programming. That way, any work done at one location can be brought to the other by simply bringing a disk along. If the home and school machines are different brands, their software disks will not be mutually compatible. This incompatibility is an unfortunate result of many computer companies trying to do something unique, and thereby eliminating the chance for a software standard all computers could adhere to. Compatibility between your computer and the software you want to buy is something you have to watch for constantly.

As good an educational aid as the computer is, however, it doesn't automatically turn every child who sits in front of it into a whiz kid. When computers are used as tutors in schools, they are used *in conjunction with* the teacher's regular coursework. The computer is not intended to replace the teacher, but rather to give each student a chance to practice skills at his or her own pace. At home, it is even more important that the student be supervised by a parent, or the computer will in all likelihood be used more for shooting aliens from the planet Gorgonzola than for learning fine points of grammar. Not every child takes to computers, just as not every child is fond of studying music.

Other Uses

You don't, however, need a houseful of computer kids to find productive applications for a personal computer. Here's a rule to follow when deciding whether you could use a computer. Take a close look at all the things you do manually—those chores you do with a pencil and paper, calculator, typewriter, and by hand. Do you work with budget and forecast figures? Do you occasionally type papers or reports? Is a lot of your time spent searching for information in your files? Do you wish you could illustrate some of your reports with professional-looking charts and graphs? Are you in charge of an organization's mailing list and newsletter? Do you do a lot of electronic banking with an automatic teller? Do you spend much research time in libraries? Would you prefer frequent updatings of the value of your investment holdings? These are only a few examples of the kinds of tasks a personal computer can perform more quickly and accurately for you. By inserting disks that turn the computer into an intelligent typewriter, a heavy-duty number cruncher, an electronic filer, a telecommunicator, and an electronic artist, you can be more efficient with these work, student, and home management jobs as needed.

Of course, you have to match the computer to the kind of jobs you intend for it. You wouldn't go out and buy a child's carpentry toy set as a first step to learning carpentry. The tools simply won't let you perform the kinds of carpentry tasks you envision. Yet so often, a seeker of computer knowledge rushes out and buys the cheapest home computer on the market with the expectation that the machine will produce a self-made computer genius. It doesn't take long before the new owner becomes discouraged with the lack of features

built into the bare console. If the original spurt of interest lasts, the user discovers that he or she must add more accessories (costing much more than the starting console) just to make a moderately functional computer. More often than not, however, the starter computer console winds up being used as a video-game machine or a doorstop. Be realistic in your expectations, and buy a computer suited for those expectations.

WHAT A COMPUTER DOESN'T DO

While the personal computer is capable of a lot of useful things, there are a few jobs around the house—jobs that seem to be mentioned too often in advertising—for which a computer is overkill. Many households don't have enough in the way of manual tasks to keep a computer busy. For example, the typical household checkbook has too few entries each month to make it worth the effort to type entries into the computer for the monthly checkbook-balancing act. A simple pocket calculator will do quite nicely.

The idea of keeping recipes on computer is one of the more bizarre home applications dreamt up, no doubt, by an advertising copywriter in search of an application for an early-model home computer. How many housewives do you know who would bother typing all the recipes saved from newspapers and magazines into a computer? Isn't it faster—and more satisfying—to clip a recipe with scissors and paste it to an index card?

Someday, recipes on computers will make sense. If you could buy a popular and respected cookbook with

recipes on a disk, you could have the computer search through the index for meals made from major ingredients on hand, based on dietetic needs (e.g., low sodium), then have the computer calculate the ingredients and amounts to feed the five people scheduled to be at your table that evening. Even so, you'll still want the recipe book with color photographs of what the final meal looks like. There's no way an inexpensive computer can re-create that visual sensation.

LEARNING ABOUT COMPUTERS WITH THE APPLE //c

You may have noticed that up until now I've said a lot about the kinds of productive things you can and cannot do with a computer, yet nowhere have I mentioned programming a computer. There is a common misconception among those who have never been introduced to computers that a prerequisite to using a computer is a knowledge of computer programming.

I am here to tell you that you can be a computer literate without knowing the first thing about programming.

A few years ago, this would have been a heretical statement. But much has changed since then to remove programming from the literacy criteria list. The concept of computer literacy has changed, as well.

Attaining computer literacy means that you can use a personal computer for a productive task—virtually any kind of application other than games. A few years ago, there was so little quality software on the market that you had to write programs to get material that

really helped you. Since then, so much prewritten software has become available that practically every kind of manual task can be run on computer without having to write the program.

Not all the software available is of high quality, mind you. Nor is it always so easy to use. Some computer manuals are as cryptic as World War II codebooks. And many programs require that you remember a set of commands in order to make the program perform various functions. In working with a couple of prewritten productivity programs, you learn a great deal about the powers and limitations of computers, as well as the quirks of keyboard commands, disk drive characteristics, and the ways computers "think." You make mistakes along the way that teach you things like making sure you have a backup copy of your valuable disks, and that you store information on a disk before turning off the computer. It's rare that anyone will ever call upon you as an experienced computer user to program a computer, but they will expect you to help them get over a snag in a prewritten program. The experience you gain in mastering a couple of productivity programs equips you with the knowledge to decode a friend's first computer manual as well as new programs that you buy for yourself. I like to equate this level of computer literacy with a high-school education, since both arm you with basic skills to help you survive the "jungle" out there.

The Apple *II*c is one of the best personal computers for gaining the experience necessary for a computer literacy diploma. It is compact, which gives you a psychological edge that you might not have over a computer that physically overwhelms you. When you first open

the *//c* box, you are supplied with a series of disks that escort you through the first stages of computer literacy. Most important, however, the library of software available for the machine is probably the largest available for any computer. Your choices in applications will satisfy just about every job you can imagine for a computer. This will hasten your arrival at computer literacy because you will have the proper software at your disposal now, instead of having to wait for the application to be written. At the same time, more and more sophisticated programs are still under development for the Apple *//c*. As a result, there will be new applications to try as your computing needs grow. And, if you like, you can do graduate work in computer literacy by getting involved with programming in many different languages (I'll have more to say about programming in Chapter 12).

In Part I of this book, I'll be describing the Apple *//c* in detail, along with an explanation of its features, both the obvious ones and what's hidden inside. Chapter 3 details all the accessories (peripherals) available from Apple for the *//c*, when you should add them, how they hook up, and what you use them for. Chapter 4 answers your questions about how the *//c* compares against other personal computers currently on the market to help you make your decision.

Part II is devoted entirely to the wide variety of things you can do with your *//c*. You'll see exactly what it's like to write a high-school essay in front of a word-processing computer, and how to use the *//c* to assist in research and organization. You'll understand what a spreadsheet and data base are all about. I'll also show

you how the *II*c can unlock artistic talent you never knew you had.

Actually, the Apple *II*c is a computer that will unlock a lot of hidden talent, and not only in the artistic sense. You'll see that even though you may think you have only one application for a computer, it will prompt you to uncover many more things you can do that will make you more productive than ever.

PART I

What the Apple //c Is

CHAPTER 1

Outside the Littlest Apple

Ask today's Apple *II* computer owners if they would like their computers to be more portable so they could take them from office to home, from home to school, from family room to den, or even on vacation, and they'd reply most affirmatively. Ask the same group if they'd rather have a disk drive built into the console instead of having it hang on as an accessory, and the ayes would have it. Ask newcomers to the Apple *II* family of computers if they'd rather have all the plug-in adapter boards already installed for them so they could attach a printer, telephone modem, and disk drive without digging into the computer's insides, and they'd bow in your direction once daily.

Put all those features together, add several Apple *II*e options as standard equipment, and you have the Apple *II*c, Apple Computer's latest and littlest machine in the tradition of the original Apple *II* computer. The *II*c is not a replacement for the popular *II*e, but rather a more self-contained, compact, and complete personal computer appealing to nontechnical consumers.

The Apple *II*c is a personal computer designed for home, student, and light-duty professional applications. It is small—slightly larger than a notebook—and lightweight so it can be easily carried under your arm in its carrying case, or by its built-in carrying handle. Among its obvious features are a full-size typewriter-style keyboard, one built-in 5¼-inch floppy disk drive, and a full complement of connectors on the rear panel that allow easy connection of Apple *II*c accessories: game controllers, a mouse pointing device, a telephone modem, a color or monochrome video display, a second disk drive, printers, a plotter, and a portable flat panel display (late 1984). All you need to add to the basic console is a video display of some kind (everything is included in the basic *II*c package for attachment to a home color TV or most computer monitors), and you're off and running. The *II*c also comes with five software disks featuring previews of many kinds of software and programming languages, as well as interactive tutorials about the operation of the computer.

Less obvious are a number of features hidden inside the cabinet. We'll be looking at them more closely later in Chapter 2, but for those who are already familiar with the terms, the *II*c has 128 kilobytes of Random Access Memory (RAM) and 16 kilobytes of Read Only Memory (ROM). The ROM contains, among other more technical things, the Applesoft BASIC programming language. You have your choice of either 40- or 80-column video display (the number of text characters displayed across the screen), depending on the kind of work you'll be doing and the quality of video monitor you'll be using. The computer operates around a new version of the original Apple *II*'s

microprocessor. The new chip, called the 65C02, is an 8-bit microprocessor built with CMOS (Complimentary Metal-Oxide Semiconductor) technology for lower power consumption and cooler operation. The 63-key keyboard is switchable between the popular Sholes (QWERTY) layout and the efficient Dvorak layout.

Now that we've seen the basic specifications, let's see what each of them means to a *//c* user.

WHAT'S IN THE APPLE *//c* PACKAGE

Unlike many other computers with similar capabilities, the Apple *//c* doesn't require an extensive setup procedure, nor does the system have to be assembled at the store by a trained individual who must then

***Figure 1-1*. The Apple *//c* comes with cables and adapters for attaching a TV or monitor, plus three manuals and five software disks.**

check it out thoroughly to make sure everything is plugged together properly. The *II*c comes packaged with everything you need to get it running except the video display.

The computer console is entirely self-contained and requires no installation of additional circuit boards. Owners of earlier Apple *II* computers will notice that the *II*c computer cabinet doesn't have a pop-off cover exposing the main circuit board. While Apple *II*e owners must add extra adapter boards to the main circuit board in order to connect most accessories, the most popular adapter boards are already built into the *II*c and accessed through a full complement of connectors along the rear panel.

A separate power transformer, called the external Floor Mount power supply, plugs into one (and only one) of the rear panel connectors. If you want to use the computer right away with a color television, all necessary components are supplied in the box: RF modulator, video cable, and antenna switchbox. If, on the other hand, you have a special computer video monitor all set up, you can use just the video cable supplied.

Packaged with the *II*c are three manuals. The first, called *Setting Up Your Apple IIc*, shows you with clear pictures how to attach the video displays and whatever extra accessories you may have purchased with your system. The second manual is called *An Interactive Owner's Guide*, and serves both as a workbook to four of the five software disks packed in the box and as a reference manual. It shows with pictures and words

how to turn on the computer, how to handle floppy disks, and what the controls on the *//*c mean.

Within fifteen or twenty minutes after opening the *//*c box, you can start using the disks, which demonstrate the powers of the *//*c and some of its software. These disks are *interactive,* meaning they not only ask you to type things in on the keyboard, but they also provide hints and other help on the screen if you seem to be losing your way. Here's what the disks contain:

Disk 1, Side 1: An Introduction. Shows you what some of the special keys on the *//*c keyboard do.

Disk 1, Side 2: The Apple at Play. Samples of several activities including Apple 21 (a blackjack game), Financial Tools (mortgage, loan, and depreciation calculations), Lemonade (a business simulation game), Music Recital (demonstration of the *//*c's sound capabilities), Space Quarks (a space shoot-'em-up), and a quiz about what you learned from various disks in the series.

Disk 2, Side 1: The Apple at Work. A demonstration of how the AppleWorks integrated program (combining word processing, spreadsheet, and data base) works. This side of the disk shows it in 40-column display format in case you're using a television set or low-quality monitor, neither of which is capable of showing the program's true 80-column display.

Disk 2, Side 2: The Apple at Work. The same demonstrations, but in an 80-column display mode.

Disk 3, Side 1: The Inside Story. A computer version of a short story, in which the dialog and pictures appearing on the screen describe how information is stored and manipulated inside the *II*c. Written in the style of a Mickey Spillane story, it also introduces the "viewer" to good practices like frequently saving on disk the work you type into the computer.

Disk 3, Side 2: Exploring Apple Logo. A demonstration of the Logo programming language, which actually has you writing a Logo program before you're through. This is an excellent example of how simple the Logo language can be.

Disk 4, Side 1 (Side 2 is blank on this disk): Getting Down to BASIC. A demonstration of the BASIC programming language. You type a few lines of BASIC programs, and learn a couple of important commands, to give you a feeling for what a BASIC program looks like.

The last manual in the *II*c box, called *System Utilities,* works in conjunction with the fifth disk, likewise labeled System Utilities. A system utility is a special kind of program you use now and then to help you organize information stored on your disks. You use it to make backup copies of important information, for example. You also use it to prepare a blank disk so it can accept information you want to store on it. This disk arranges all your system utility choices in convenient menus, and takes care of most of the technical concerns for you. The System Utilities disk is described in more detail in Appendix D.

THE //c CONSOLE

The main console of the Apple //c contains all of the computer's circuitry, plus a full-travel, typewriter-style keyboard and one 5¼-inch disk drive, in a very small cabinet measuring 11½ × 12¼ × 2½ inches. The cabinet is an ivory-colored plastic. The keyboard and disk drive door are more like a putty color, frequently used on cabinets of professional computers like the Macintosh. On the rear panel of the //c is a hinged rubber-coated carrying handle that lies flush with the rear of the cabinet when the computer isn't being used or carried. When the computer is used, however, you are supposed to swing the handle 180 degrees until it locks in place, protruding downward from the cabinet. This serves two purposes. First of all, the handle raises the rear of the entire console to give a more pronounced slope to the keyboard, making it a bit easier to type on than if it were lying flat. Secondly, the raised console improves ventilation around its body, helping reduce heat that might build up during operation. Ventilation holes underneath the keyboard toward the front of the cabinet and along the top rear of the cabinet provide ample flow-through ventilation to keep the //c running quite cool without a fan.

Having a big hand in keeping the computer cool is the fact that the traditionally hot power transformer is kept in a separate power box, called the Floor Mount transformer, located midway along the AC power cable. Instead of the usual big, ugly black transformer box, Apple's is a white and putty affair matching the coloration of the //c cabinet. The entire cable length, from computer to AC plug, is twelve feet, long enough for just about every installation.

The *II*c is definitely portable. Weighing only seven and a half pounds, it's scarcely heavier than a college textbook. Of course, you also need the power transformer, which adds some bulk to the package. But together, the two weigh only eleven pounds, considerably lighter than so-called portable computers in the sewing-machine-sized cases, which typically weigh between twenty-five and thirty pounds. If you need to carry home the computer from the office frequently, you might want to buy an extra Floor Mount transformer, keeping one each at the office and at home, so you only have to bring your computer console with you. Likewise at home, you may use the *II*c with a color TV in the family room for game playing, and then move it to the study to write a report with a text monitor. If you have a Floor Mount transformer in both locations, it makes the move much easier, and involves plugging in only two cables (power and video display) to get going.

THE KEYBOARD

For many reasons, the Apple *II*c's keyboard is really quite remarkable. For one thing, its action represents a departure from Apple's other keyboards. Instead of having a smooth vertical action, each key has what is known as breakover action, similar to that found on many electric typewriters. Here's what breakover action means. With any keyboard key, you press the key down to a certain point where a contact inside the key closes, causing that character to be sent to the computer console. With a smooth action key, you're never quite sure when the contact closure takes place, unless you always press the key the full vertical travel distance. But what about a partial hit, when your

finger slips slightly? You can't tell by the feel, so you have to look up at the screen to see if the key registered. With a breakover key, however, you get tactile (touch through your finger) and audible (click) feedback that you have made contact. When you reach that contact point, the tension of the key gives way slightly. This kind of keyboard contact is like the one on the IBM Personal Computer keyboard, but the *//c* breakover is less severe. The result is an excellent feel for the experienced typist who may be used to the feel of an electric typewriter keyboard.

In addition to the standard typewriterlike keys, Apple incorporates a few extra keys on the *//c* keyboard, as it does on the *//e* keyboard (the layouts are identical). At the upper left corner is a key labeled ESC for escape. The function of this key varies from program to program, but it usually is the one you press when you want to get back to the beginning of a program. Three keys—Control, Open Apple, and Closed Apple (the last two are located on either side of the space bar)—operate like extra Shift keys. You hold one of them down and press another key on the keyboard to issue a command, depending on the program you're using at the moment. Often, such commands include the first letter of the corresponding English command (e.g., Open Apple-P for Print; Open Apple-C for Copy). Four cursor control keys (the cursor is the flashing box on the screen where the next character you type will appear) are located at the lower right corner of the keyboard. The Return key, along the right side of the keyboard, is one of the most frequently used keys of them all. Whenever the computer asks you to type information in response to a question, you need to press the Return key to send your response to the

computer. That's why, on some computers, this same key is labeled Enter. Oddly enough, while typing in a word-processing program, the Return key is rarely used the way it would be on a typewriter. At the end of each line, the computer usually knows when a word won't fit in the margin and automatically starts it on the next line. The Return key is used, however, to signal the end of a paragraph.

One thing that makes the *//c* keyboard unique is that it is the first on a popularly priced personal computer to offer the user the option of switching at a moment's notice between the Sholes keyboard layout (better known as QWERTY for the first six letters across the top row of letter keys) and the Dvorak keyboard by simply pressing a button. Briefly, the QWERTY layout for keys was designed back in the earliest days of mechanical typewriters. Inventors discovered that some typists could strike the keys faster than the key mechanisms could respond, jamming the machine. To compensate for deficiencies in the mechanics, the designers deliberately located letters on awkwardly arranged keys to slow down the typists. That layout remains the standard today.

But recently, there is growing awareness of a more efficient layout, called the Dvorak keyboard, after its inventor, Agustin Dvorak. The keys are laid out according to the most often used key sequences placed on fingers of alternate hands. The most common letters are located along the "home" keys, reducing the amount of finger movement necessary for most words. Compare the layouts of the QWERTY and Dvorak keyboards in Figure 1-2.

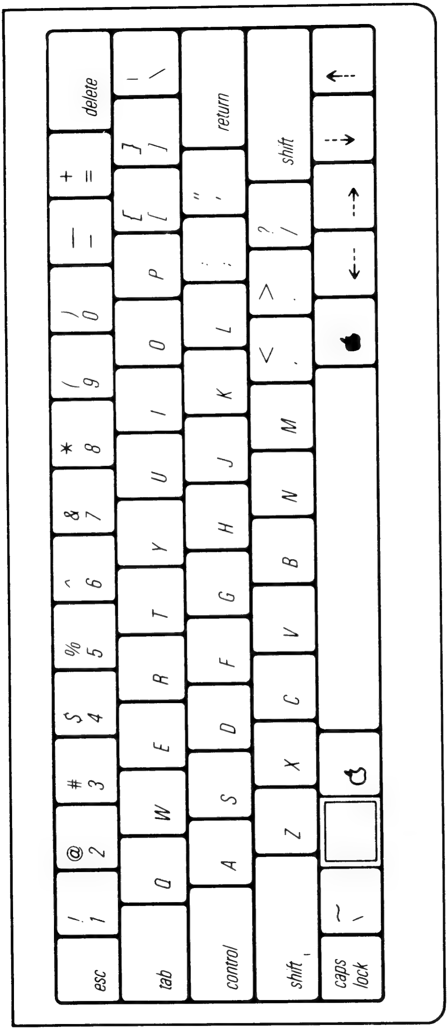
Keyboards on a few other computers can be converted to Dvorak keyboards with special software. But to select between QWERTY and Dvorak keyboards on the Apple *II*c, all you do is press one of the slim buttons above the keyboard with a pencil or other semisharp object. It helps to know the Dvorak layout, however, because the letters for the QWERTY keys are the only ones on the keycaps. If you are a dedicated Dvorak keyboard enthusiast, however, you can pop off each of the keys and reinstall them on the keyboard according to the Dvorak layout. Later in 1984, Apple will offer a set of Dvorak keycaps that fit over the existing keys.

One feature of the keyboard design should put to rest parents' fears about young kids playing with the *II*c. Beneath the keys is a layer of plastic that prevents liquids from getting down into the key contacts and other circuitry of the computer. When the computer is propped up on its handle, any liquid that is accidentally spilled on the keyboard is directed out the front air vents. The Apple designers call this their "drool-proof" keyboard.

INTERNAL DISK DRIVE

A computer disk drive is in some ways analogous to the turntable of a stereo system. Both play software disks that bring the hardware to life. A computer disk drive, however, also records information onto disks. And it's safe to say that a computer disk drive is a much higher precision instrument than a turntable.

At the right rear side of the *II*c cabinet is a slit into which a disk is inserted. A latch, or door, guards the



esc
tab
control
shift
caps lock
delete
return
shift

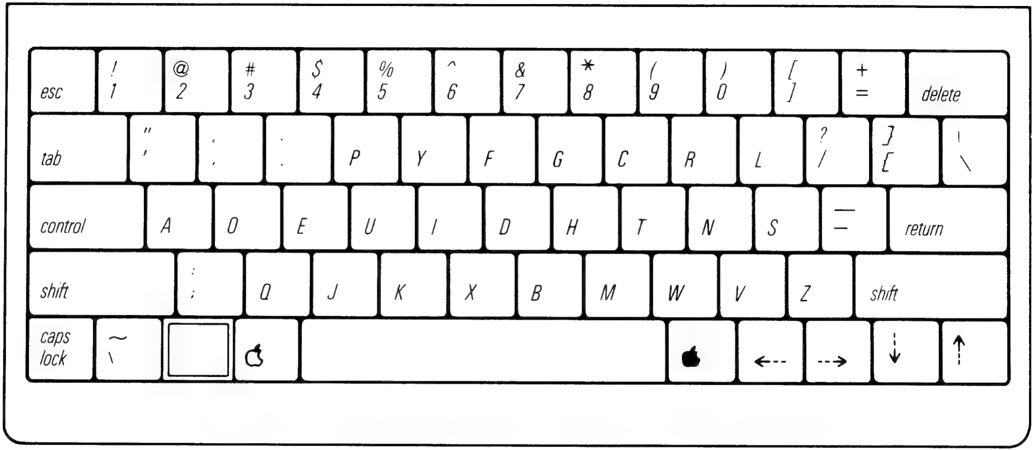


Figure 1-2. **The standard Sholes (QWERTY) keyboard layout (top) and Dvorak alternate keyboard arrangement (bottom) built into the //c.**

slit so you can't accidentally remove a disk while it is being used. The door on the *//*c disk drive is a design unique to Apple. If you have previous experience with computer disk drives, you might not be prepared for this one.

While most disk drive doors lift up or rotate, the *//*c door pops up only after you *press inward* on the latch. If a disk is already in the drive when you pop the latch, the disk is partially ejected so you can grip it easily for removal. When you insert a disk into the open drive, it is grabbed lightly inside and held in place until you push down on the latch, securing the disk in place. This latch design was chosen because of its ease of use when approached both from the side (as on the console) and from the front (as with an external disk drive).

The actual disk medium is a round sheet of pliable plastic (usually Mylar) that is coated with microscopic metallic oxide particles whose arrangement on the disk is highly susceptible to alteration by a magnetic field. The material is very similar to that used on audio tape in cassettes. A disk, therefore, might be likened to a series of concentric strips of recording tape. Each disk is housed in a flexible plastic sleeve which protects the surface of the disk from being touched. Holes in the sleeve allow the disk drive to transfer information on and off the disk (see Figure 1-3).

Like a cassette player/recorder, a disk drive has a read/write head that comes into contact with the disk to transfer information. When recording information on a disk, the write head sets up a strong magnetic field on a very precise spot. The disk drive spins the disk at

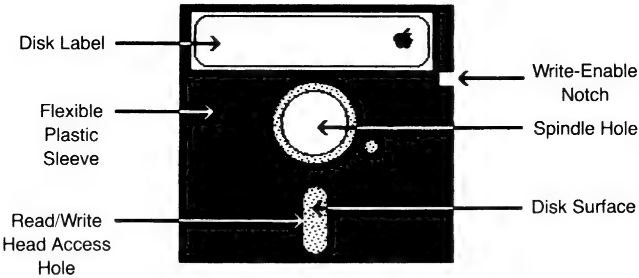


Figure 1-3. **Parts of a floppy disk.**

a speed of 300 revolutions per minute (rpm) beneath the write head. The write head's only movement is in a straight line along the long hole in the sleeve. Information from the computer is converted into specific magnetic field pulses on the head, which set the metallic particles on the spinning disk's surface in a fixed pattern. When it comes time to read that information from the disk, the read head sets up a smaller magnetic field which detects the patterns of metallic particles on the disk and reconverts them into the pulses that the computer can understand.

Disks have to be handled carefully. Even though they are called floppy disks, and they are indeed pliable, they should not be bent. You run the risk that the surface of the disk might be creased, causing a considerable amount of information along that crease to become garbled or unreadable by the disk drive's read head. Moreover, the minute size of each speck of information makes it imperative that the surface of the disk remain clean. Even smoke and dust particles, though scarcely visible to the unaided eye, turn into

good-sized boulders in the path of the read/write head. Body oils are another source of trouble for a disk. Freshly washed hands still have enough oil on them to gum up the works when a read/write head makes a pass over the surface of the disk. Never touch the actual disk material. There is plenty of sleeve surface to hold the disk for insertion and removal from a disk drive. Always place a disk not in the disk drive into its protective envelope, making sure the exposed disk holes are facing downward, inside the envelope. There are other precautions, which are detailed in the *II*c owner's manual and on the back of nearly every disk envelope.

REAR PANEL CONNECTIONS

One major feature that sets the *II*c apart from the *II*e, and makes the new model much more like the Macintosh, is that the rear panel contains built-in connectors (with the corresponding adapters built inside) for just about every kind of accessory (peripheral) you are likely to want to add to the computer. Several of these rear panel connectors, called D-type connectors, accept matching cables which are fastened with a pair of hand-tightened screws for easy attachment and removal when needed (on other computers, connectors must often be fastened with a screwdriver). Each of the rear panel D-type connectors is of a different size, so there is no chance you can plug a cable into the wrong socket and do damage to the computer or peripheral. Other connections (except the one for a video monitor) use circular connectors, called DIN connectors, a popular standard pioneered by European electronics manufacturers.

Also facilitating plugging in the right cable to the right connector is a pictograph of the appropriate peripheral above each connector, with the same picture appearing on the cable connector. Simply match up the pictures. You won't confuse the printer connector with the modem connector (which are identical five-pin DIN plugs), since the former is associated with a picture of a printer, while the latter shows a telephone handset. In fact, these pictures are universal enough to make them valid in any country, regardless of the principal language.

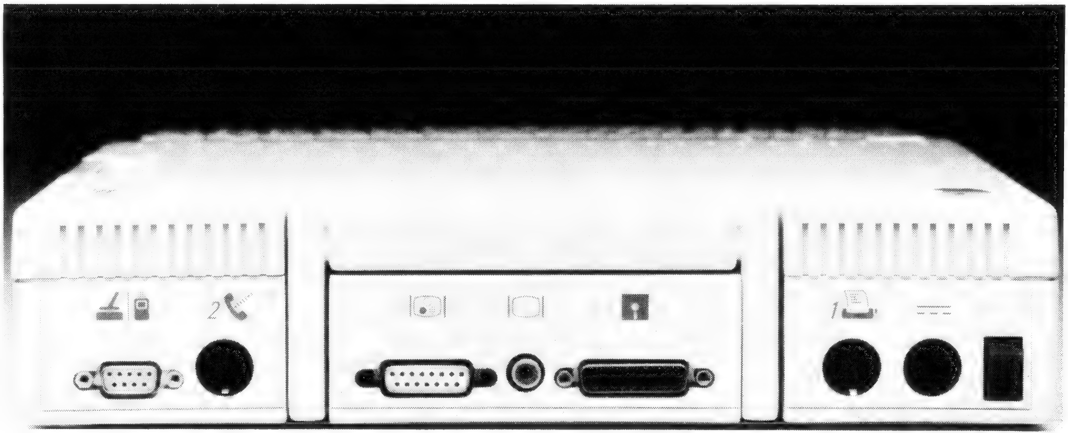
We'll be examining the peripherals in full detail in Chapter 3, but let's see what the *II*c is ready to accept.

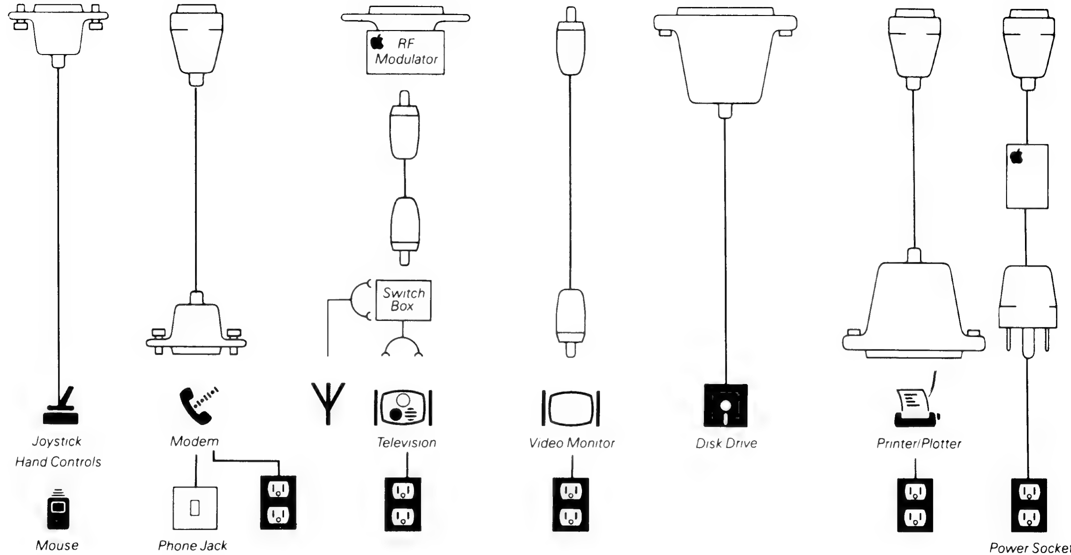
Starting on the left (when looking at the rear panel), the first connection is for game controllers or a mouse pointing device. While the use of game controllers on the *II*c is obvious, the use of a mouse may be new to you. Some new software programs use a mouse as a way of pointing to things on the screen, thereby communicating with the computer (much in the same way the Macintosh mouse performs one-handed commands for its user). We'll see an excellent example in Chapter 9.

The second connector from the left is for a telephone modem. A modem converts computer signals into audible signals that can be transmitted over a telephone line. If you become involved with any applications in Chapter 8, a modem will play a critical role.

Next is a connector for displaying *II*c output on three types of displays: a color television, a flat panel display, and an RGB color monitor. Before you can

Figure 1-4. **All Apple //c peripherals attach to the console's rear panel connectors.**





Joystick
Hand Controls
Mouse
Modem
Phone Jack
RF
Modulator
Switch
Box
Television
Video Monitor
Disk Drive
Printer/Plotter
Power Socket

connect any of these displays, however, you need a corresponding accessory "docking module" connector. One of these, called an RF modulator, is packaged in the *II*c box. It converts the video output to something like a TV station signal that the TV set can receive. Also supplied with the *II*c is a switchbox, which attaches at the antenna terminals of your TV set and lets you select between watching TV or using the set as a computer display. Connecting the flat panel display (scheduled for late 1984) or a high-quality RGB color monitor (available from outside sources) requires a special "docking module" for each one. More information about these modules will be available later in 1984.

The little round connector in the center of the rear panel is the source of another kind of video output, composite video. This is the output you use if you hook up a special computer monitor, either monochrome or composite color (see Chapter 3 for a complete discussion of the variety of display devices for the Apple *II*c).

The large connector to the right of the video output is for the attachment of a second disk drive. You may find a second disk drive useful for some applications programs, as you'll see later. A second drive also facilitates copying information from one disk to another, such as making backup copies of important disks. Without a second drive, you may have to do some disk swapping in and out of the internal disk drive to accomplish these disk maintenance chores.

Further along the rear panel is the connector for a printer or plotter. Apple offers compatible printers for

both fast dot-matrix printing and near letter-quality printing for the professional who needs finished documents to look as though they were typewritten on a quality electric typewriter. A new low-cost Apple printer even prints in color. You would use a plotter if your applications required precisely drawn graphics output, including overhead transparencies.

The last connector, on the right, is for the power cable coming from the Floor Mount transformer. Since plugging this cable into any other rear panel connector might damage the computer, its plug fits only into the appropriate power connector. Next to the power connector on the rear panel is a rocker-type switch—the main power switch for the *//c*.

OTHER CONTROLS AND LIGHTS

There are only a few other controls and indicator lights that may not be obvious to you. Recessed under the left-hand corner of the keyboard are a knob and a small audio jack. The knob controls the volume of the audio coming through the *//c*'s built-in 1½-inch-diameter speaker, which is located near the center of the front air vents. If you're working on a game or music program and don't want to disturb others around you, you can plug headphones into the jack—a ⅛-inch jack like those used in Walkman-type stereos. In fact, the jack is wired so it sends audio to both sides of stereo headphones (although not in stereo). So, if the only pair of headphones with a ⅛-inch jack you have around are the ones from your personal stereo player, then go right ahead and use them with the *//c*. The volume control also works with output to headphones. The volume on most headphones will be rather loud at

a comfortable speaker setting, so it is best to start out with the volume control set at its lowest and adjust the knob slowly upward until it is comfortable for headphone listening.

Just above the top row of keys on the keyboard are some push-button switches and lights. Starting at the left, the large button is a reset switch, which you use in conjunction with two other keys (Control and Open Apple) to reset the computer when you want to start up a new disk program. Resetting the computer this way erases anything that is currently in memory. With such an awkward three-key simultaneous sequence, it is essentially impossible to reset the computer unintentionally.

Two slim push buttons to the right of the reset switch aren't buttons you'll use often, so they require a reasonably sharp object, like a pencil point or fingernail, to press down. One switches back and forth between the QWERTY and Dvorak keyboards. The other one is used whenever you switch from a video monitor to a television set and controls whether you are in 40- or 80-column-wide text display. The action from this switch is not instantaneous like that of the QWERTY/Dvorak switch. You must set this switch before you start up a program.

On the right side are two indicator lights. The green one is a power-on light. It shines as long as the computer is turned on. That way, even if you turn off the video display, you'll be reminded that the computer is still left on. The other light glows red whenever the disk drive is in action. When the red light is on, it is signaling you to not disturb the disk drive,

otherwise you might garble information stored on the disk.

That's all there is to the *//*c's switches and connectors. The plot thickens, however, when we go inside the computer and learn what some of the more technical specifications mean to a *//*c user.

CHAPTER 2

Inside the Littlest Apple

Not everyone needs to know about the "innards" of a computer. But if you're doing comparison shopping for a personal computer, you might as well learn what all the jargon and specifications mean. Here's a look under the Apple *II*c hood.

THE MICROPROCESSOR

The vast majority of the circuitry that makes the Apple *II*c run is contained in a few dozen little black rectangular cases, called integrated circuits (ICs). They are mounted on a large circuit board that is just about as big as the *II*c case (see Figure 2-1). ICs are more commonly called chips, yet the actual chip, a quarter-inch square of silicon, is housed inside the black, many-legged cases we see. These black cases are as large as they are for the convenience of humans who handle them and the machines that insert and solder them to circuit boards in high-speed assembly processes. The tiny chip inside the black case can contain the equivalent of many hundreds of thousands of

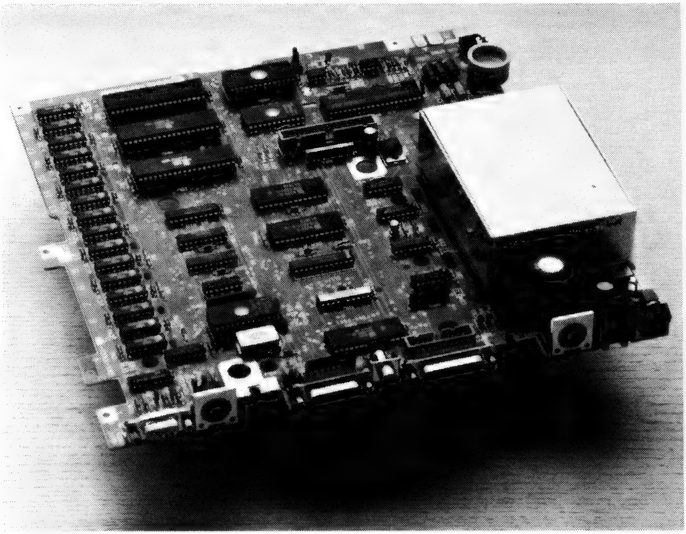


***Figure 2-1*. All the "computing" takes place on the main logic board inside the //c cabinet.**

transistors and other electronic components. When computer popularizers say that a fingernail-sized sliver of silicon contains the same computing power as the room-sized computers of the 1940s, they're not kidding.

There is one chip in particular that keeps all the other ones in order. It coordinates the actions of the chips and keeps tabs on everything going on inside the computer. It's called the microprocessor. While personal computers currently on the market use hundreds of different chips for the various parts of their circuits, it seems that manufacturers rally around only a few microprocessor models. The names of the chips are

not fancy marketing names, like the computers they run, but rather part numbers designated by their designers. If you've ever thumbed through computer literature, you've probably seen specifications list the same few chips over and over. The most popular ones today (and their original manufacturers) are the 6502 (MOS Technology), the Z-80 (Zilog), the 8080 (Intel), the 8088 (Intel), and the 68000 (Motorola).

The microprocessor used in the Apple *//c* is a derivative of the 6502, the chip used in the original Apple II. It is called the 65C02, where the *C* stands for CMOS (Complimentary Metal-Oxide Semiconductor), a special manufacturing process that produces a chip with very low power consumption. One reason the 65C02 was a good choice for the *//c* is that since it is a close relative of the original Apple II chip, most software written for other members of the Apple *//* family will run on the new chip with little or no modification. Therefore, you don't have to wait for a large software library to be developed or converted to run on a *//c*. You'll be able to draw from a huge existing software library.

There is a lot of talk today about microprocessors being 8-bit, 16-bit, or 32-bit. Since the 65C02 is an 8-bit microprocessor, and therefore seems to be at the low end of the pecking order, I should explain what the whole thing means. To do so, however, requires a little side trip into the world of computer "bits."

Running through the veins and arteries of the computer circuits (and to peripherals outside the computer, too) is a lifeblood consisting of electrical pulses. The theories behind this flow and how every chip

knows what to do when it receives a certain stream of pulses are outside the scope of this book. Besides, it's not easy to grasp much of what happens at chip level, because the number of components crammed into a tiny chip and the speed at which the pulses race about the system stretch the imagination. Let's face it, anything that happens at the rate of millions of times per second creates little more than a blur of a mental image. But that's how fast electrical pulses are flowing through the *//*c.

These pulses are really nothing more than electrical states—either the presence or absence of current—racing through the chips and from chip to chip. Integrated circuits are designed to respond in very specific ways each time they receive a predetermined sequence of current states. For human convenience, these electrical states are more commonly referred to as "on" and "off" or "1" and "0." Each 1 or 0 that goes through the circuit is called a bit, a contraction of the technical term BInary digiT.

Binary math uses only the numerals 0 and 1, while the decimal system you're more familiar with counts from 0 through 9 before carrying one digit to the left. And when you carry a digit to the left in binary, the number increases by a factor of two (ones, twos, fours, eights, sixteens, etc.) instead of the decimal system's factor of ten (ones, tens, hundreds, thousands, etc.). For example, see how the decimal number 225 is counted in both decimal and binary math in Figure 2-2. The Apple *//*c's inner circuits use such binary numbers as representations of our alphabet and other common characters. All together, these binary numbers form a kind of language linking the electronic components together.

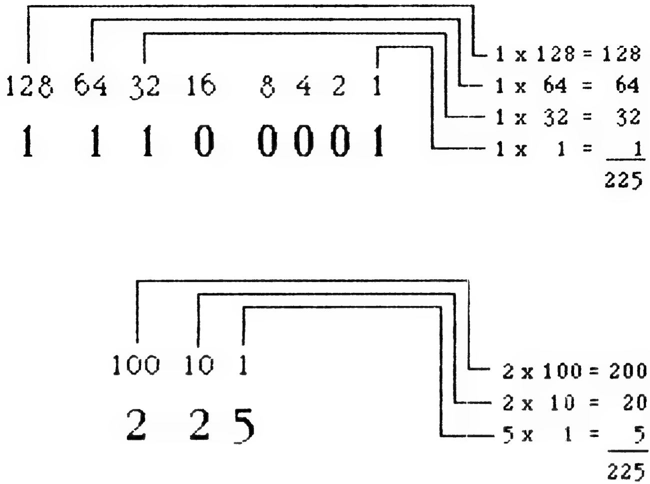


Figure 2-2. **The number 225 portrayed in binary and decimal numbering systems.**

In computers like the Apple *//*c, Commodore 64, and Atari 600XL/800XL, information moves around the computers eight bits at a time. In other words, 8 separate on/off states, or "data lines," run side by side through the microprocessor and other chips. These computers are said to run with an 8-bit microprocessor (CPU chip). In computers like the IBM PCjr, the Intel 8088 microprocessor handles information 16 bits at a time inside the chip. In communicating with the rest of the system, however, the 8088 works in 8-bit-wide paths. Even though the 8088 is really an 8- and 16-bit hybrid microprocessor, it is nonetheless called a 16-bit chip. Similarly, the Motorola 68000 microprocessor, used in the Macintosh and Lisa computers, works internally 32 bits at a time, while communicating with

other chips in 16-bit chunks. Apple calls both computers 32-bit computers, even though their microprocessors are actually operating as 16/32-bit hybrids.

Generally speaking, computers controlled by microprocessors with 16-bit-wide internal paths perform math calculations faster and can manage more computer memory at one time than those with only 8-bit-wide internal paths. A similar improvement occurs when advancing from 16- to 32-bit-wide paths inside a CPU. The question at hand, however, is whether the Apple *//c*'s 8-bit microprocessor has enough power and speed for the kinds of things you want to do with a computer.

Although a lot of computer programming attention is currently directed at newer 16- and 32-bit technology, the capabilities of 8-bit microprocessors have yet to be fully exploited. In Part II, you'll see some examples of remarkable 8-bit software recently developed for, or adapted to, the Apple *//c*. Often, the inspiration for an advanced 8-bit program comes from something developed for a more sophisticated computer. Experienced programmers, however, are tailoring those complex programs for the 8-bit environment, providing *//c* owners with a continuing influx of exciting state-of-the-art software.

MEMORY

Before we discuss how much and what kinds of memory the Apple *//c* holds, let's look at the basics of computer memory, using a sheet of graph paper as an example. Instead of plotting dots or graphs on the paper, however, we'll be using the boxes formed by

the intersecting lines. We'll be filling boxes with characters like letters of the alphabet, numbers, and punctuation symbols. But first we need to go back to our discussion about bits for just a moment.

Remember we said that the computer only knows how to shuffle 1s and 0s—bits—around the circuits. Whenever the letter *A*, for example, is typed on the keyboard, the computer must convert it into a predetermined code composed of a sequence of bits. Later, that same sequence may be sent to the video display circuit where it is converted back into the *A* we recognize.

Accommodating all the letters (upper- and lowercase), numbers (0–9), and common punctuation (!"#$%&'()*+,−./:;<=>?@[\ ]^_`{¦}˜) in a sequence of 1s and 0s requires a minimum of 96 unique bit sequences. Some years ago, a standards committee agreed to a table of bit sequence codes for all 96 characters, plus over a dozen special control signals used inside the computer and with printers. A total of 128 different sequences was assigned by the American Standards Committee for Information Interchange (ASCII, pronounced *ass'-kee*).

To represent all 128 sequences requires using all the possible 7-digit binary numbers (0000000 through 1111111). But the standards group set the standard code at 8 digits long, allowing communication of as many as 256 different characters. Here are examples of 8-digit binary numbers, their decimal equivalents, and standard ASCII characters they represent:

Binary	*Decimal*	*ASCII Character*
00110011	51	3
00111010	58	:
01000001	65	A
01110101	117	u
11001000	200	not assigned

A special term is used to describe an entire 8-bit group, shown in the Binary column above: the byte (pronounced *bite*). One byte, therefore, is the amount of computer information (1s and 0s) needed to convey the meaning of a single character. Now back to our graph paper.

Each square on the graph paper represents the space taken up by one character, or one byte. In personal computers like the *//c*, memory is most often counted in thousands of bytes, or kilobytes. To be technically accurate, a kilobyte is precisely 1024 bytes—due to a quirk of binary arithmetic—but it's easier to think of a kilobyte as an even 1000. A kilobyte is most often abbreviated with the simple letter *K*. A computer rated as having 64K, then, has room in its memory for about 64,000 (65,536 to be exact) bytes, or characters.

When you turn on the computer, one kind of memory inside the computer, called Random Access Memory (RAM), looks as empty as the blank sheet of graph paper. Every time you press a key on the keyboard, the byte representing that character goes into one of the squares. But even blank RAM really isn't as featureless as it seems, since every square has a number serving as a kind of location number (address) for that box. This is important, since the microprocessor must keep track of where everything is stored in RAM by remembering addresses.

*II*c RAM

On the Apple *II*c Main Logic Board—called the motherboard by computer techs—are 16 memory chips which make up the 128-kilobyte RAM of the *II*c. Why 16 chips?

The current state-of-the-art in affordable RAM chips used in personal computers is at the 64-kilo*bit* level—each chip can store roughly 64,000 bits, or 8 kilobytes. Therefore, to store 128 kilobytes, you need 16 chips ($16 \times 8K = 128K$).

RAM is really a misnomer because there are other kinds of memories that can be read by a microprocessor at random just by knowing the address of a batch of characters. More accurately, what we call RAM should be called Read/Write Memory, because its most important attribute is that new information can be written into the memory cells while the computer is running.

Of all of RAM's properties, however, the one you should remember every time you use the *II*c is that RAM does not permanently store information. In the *II*c, the power must be on to preserve anything stored in RAM. The minute you turn off the power, whatever was in RAM is lost. It's as though you took your filled-out graph paper, tore off the top sheet, and tossed it in the fire. The next time you turn on your computer, you get a completely empty, fresh sheet of paper.

While this may sound dreadful, most programs you'll be using will have safeguards built in to remind you to save your work in RAM on a more permanent storage medium—a floppy disk. But if the worst should hap-

pen to you, welcome to the club. Losing your data by prematurely turning off the computer is an initiation ritual that every new computer user goes through at least once.

Not all memory is so volatile, however.

//c ROM

Another kind of memory, called Read Only Memory (ROM), comes with characters already imbedded in it, as if the graph paper were to come out of the package with characters already printed in all the squares, with no room for you to write in your own characters. At the chip factory, *//c* ROMs are built to contain instructions written by *//c* engineers. You can't change what is designed into a ROM. Like a RAM, the contents of ROM are stored one byte to a box (in the graph paper analogy) and can be read at random simply by knowing the address.

What does the ROM contain? All computers need some kind of instructions to get them operating on the most rudimentary level—like knowing what to do with a character when a keyboard key is pressed, how to display the character on the video monitor, and how to start getting information from a disk inserted into the drive. These instructions are usually called firmware because, as part of the basic computer, they are hardware, yet the instructions themselves are actually written as software. Without this firmware, the chips of the *//c* (or any other computer) would just lie there in a useless heap.

The Apple *//c*'s ROM is 16 kilobytes big. It contains a version of the BASIC programming language called

Applesoft BASIC, some special routines that advanced programmers use, and a number of other instructions that keep the various parts of the system working together.

How the *II*c Uses RAM and ROM

For the computer to follow instructions from a software program, the 65C02 microprocessor must be able to find its instructions from someplace within the *II*c's RAM or ROM. Thus, with disk programs, you must literally transfer a copy of the instructions from the disk to RAM-type memory. The process is called *loading the program,* and is usually automatic when you turn on the computer with a disk in the drive (although it sometimes involves pressing a few keyboard keys). Every program comes with precise loading instructions anyone can follow. When you load a program from disk, you are simply loading a copy—the original program is intact, ready for the next time you want to use it.

A program loaded into RAM from disk rarely uses up all the available RAM in the computer. In fact, if the program is supposed to work with words or numbers you type in, then there must be enough RAM space to accommodate your *input*.

Everything you type into the computer goes first to RAM. But since the contents of RAM are erased when you turn off the computer, you will probably want to save your input (a word-processing document, figures for a monthly budget, etc.) on a disk. This process is called *saving* to disk. At some later time, when you wish to work with your document or figures, you first

load the appropriate program into RAM and then load in the other data.

VIDEO OUTPUT

The Apple *//*c, like many personal computers capable of color graphics, features several different video output modes, which are controlled by the programs you run or write. The two basic video mode categories you'll be dealing with are text (sometimes called alphanumeric, since the screen characters are letters, numbers, punctuation) and graphics.

Text Modes

Text mode is what you use in most professional applications when working primarily with words and numbers. Most of the time, you will want to see as much of a document, spreadsheet, or form in one screenful as possible. Since a typical typewritten document has margins allowing about 65 characters across a page, it would be most convenient in programs like word processing to have a display mode that lets you see at least that many characters across the width of the screen.

The *//*c has two text modes, 40 and 80 columns wide. Both display 24 lines of text. The 40-column width produces large letters on the screen, which some low-cost color monitors and all television sets need in order to display readable text. More sophisticated monitors (predominantly monochrome—black-and-white, black-and-green, black-and-amber—for the *//*c) allow sharp display of 80 characters across the screen. Occasionally, a software designer will predetermine that a 40-column display is better for a particular

program (as when text and graphics are to be mixed on-screen), and the program will preset the display mode accordingly. But if you are using a productivity program designed for 80 characters across the screen, such as a spreadsheet or telecommunications, you should choose the 80-column text mode by placing the text mode switch (just above the *//c*'s keyboard) in the 80-column position before you start the program. Bear in mind, however, that the video display you have hooked up to your computer must be capable of 80-column video for you to be able to read the text plainly.

The *//c* makes effective use of its built-in 80-column text mode by including 32 graphics characters. A graphics character in text mode differs from a shape formed in the graphics mode. A graphics character in text mode is predesigned by the computer's engineers, and is no larger on the screen than the area taken up by a single letter or number. A graphics shape in graphics mode, on the other hand, can be created by the user, given attributes, and sized according to the needs of the program. Apple calls the *//c*'s text mode graphics characters Mouse Text, although you don't need an AppleMouse to use them in your programs.

Programmers are likely to use Mouse Text characters for making more visually interesting text screens with graphics borders. The symbols can also be used to create a Macintosh-like visual environment, since some of the characters have the same meanings as the Macintosh's icons (e.g., an hourglass asking you to wait while loading data from a disk) and screen structure (e.g., menu bars, close document boxes, etc.). Command lines on program screens can display a key

sequence, like Open Apple-A, with an actual Mouse Text Apple symbol (in outline form for Open Apple, solid for Closed Apple). Mouse Text will spark the visual creativity of many programmers.

Graphics Modes

Three different graphics modes on the *II*c are used for color displays, graphics shapes, and animation. What distinguishes each graphics mode are the resolution and number of colors displayable at one time.

Graphics resolution is measured by what is known as a pixel count. Pixel is a shortened expression for picture element, the smallest display unit on the screen under control of the computer. The more pixels you have on the screen, the smaller each pixel is, allowing more detail in the graphics design.

The Sunday color comics provide a good example of what picture elements and graphics resolution mean. If you look very closely at the printing of the Sunday comics or in a comic book, you'll notice that the colors and shapes consist not of continuous swaths of ink, but of tiny dots. By using small dots (picture elements), the printing process can capture fine details like facial features and smooth, curving lines as originally drawn by the cartoonist. Now imagine what the picture would look like if the dots were two or four times larger than they are now. Shapes would no longer have smooth edges, because you could readily see the individual dots. And any nuances of facial expression would have to be left out because the dot pattern is too coarse to depict them in sufficient detail.

On the 128K Apple *II*c, there are three different levels of color graphics resolution (number of pixels) avail-

able. As the resolution increases, the video display demands more information from the computer as to which picture elements are "on" and what color attribute each pixel gets. Within the old Apple II Plus limit of 64K built in (only a portion of which is available for graphics, while the rest holds the program and other data), as the density of pixels increases, the number of color options (attributes) per pixel must decrease. Thus, the more detailed the graphics on the screen, the fewer colors can be displayed at one time. By expanding memory to 128K, however, the *IIe* and *IIc* can display even higher resolution pictures with a full complement of color.

The lowest *IIc* resolution color graphics feature 40 picture elements across by 48 down. In this mode, up to 16 colors can be shown at one time. A number of educational programs use this mode, since it provides the most entertaining color. But each picture element is a rather large box on the screen, so graphics shapes are not very detailed.

The most common graphics programming is done in what Apple calls high-resolution mode, giving the programmer 280 dots across by 192 down. In this mode, up to six colors can be displayed at once, which is plenty for most educational and entertainment programs.

A mode until now not often exploited by Apple *II* programmers—double-high-resolution graphics—offers 560 × 192 pixels on a screen for very high resolution with 16 colors displayable on the screen at one time. While this graphics mode is clearly the most visually appealing, it hasn't attracted too many programmers, because the standard 64K RAM of earlier

Apple *II* models had to be shared by the video output and the program. Sixty-four kilobytes just wasn't enough memory for the programmer to generate such a finely detailed picture. Even the Apple *II*e, which is expandable to 128K, comes with only 64K as standard. Software developers like to write programs for the lowest common denominator computer, to make sure that everyone who owns an Apple *II*e can use their program. Consequently, few programmers would limit their potential market by writing only for those who had expanded their *II*e to 128K. But with the *II*c coming equipped with 128K as standard, there is a strong likelihood that more double-high-resolution graphics software will surface in the near future.

Here, then, is a summary of the Apple *II*c's video modes:

Resolution	*Mode (columns × rows)*	*Colors*
40-column text	40 × 24	b&w
80-column text	80 × 24	b&w
low-resolution graphics	40 × 48	16
high-resolution graphics	280 × 192	6
double-high-resolution graphics	560 × 192	16

Selecting a video monitor for the kind of video modes you need takes special care. In the next chapter, we will see how to match the right monitor to the right video mode for the most productive combination.

So far, you've seen the Apple *II*c inside and out, and probably picked up more knowledge about computers and jargon that you thought you'd ever have. Now we'll look at all the options available to *II*c owners with the addition of peripherals like video monitors, printers, and much more.

CHAPTER 3

Adding to the Apple *II*c

While the Apple *II*c is a capable little computer by itself, adding a few extra goodies to its rear panel connectors can make it a full-fledged workhorse of a personal computer. Printers, a mouse pointing device, and a telephone modem expand the powers of the Apple *II*c beyond its innate qualities. But the first thing you must take care of is the video display.

VIDEO DISPLAYS

Not all video displays are alike. A color television, for example, can't legibly display some text screens that come out of an Apple *II*c. A monitor that does the best job showing text won't display color graphics. In other words, one monitor does not fit all.

Generally speaking, your choice in a monitor depends largely on the kinds of things you'll be doing with the computer. If your computing consists mostly of words and numbers, you should have a monitor that is optimized to present the *II*c's 80-column-width text mode. But if your applications are almost entirely entertainment and educational, then a color monitor or home

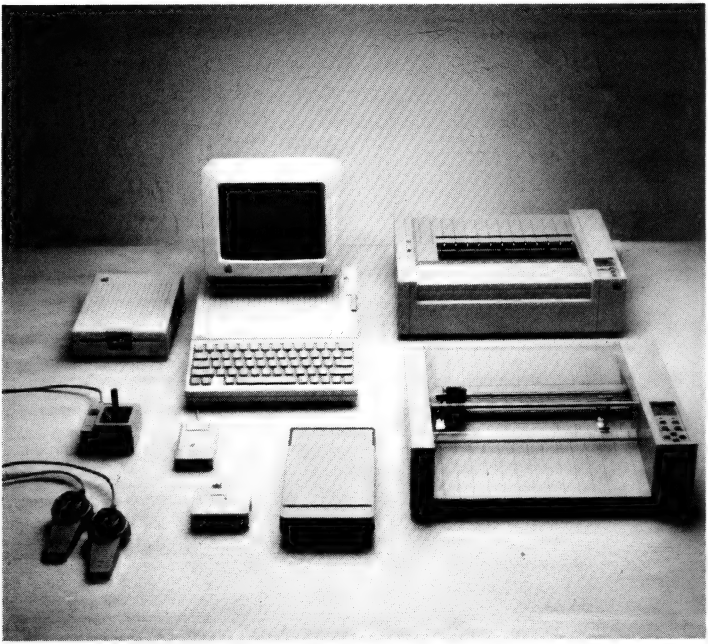


Figure 3-1. **A full range of Apple brand peripherals are available for the //c.**

television will do the job. Fortunately, the portability of the //c lets you set up the machine as an educational computer in the family room with a color TV, and then move it to the kitchen or den with a computer monitor that lets you see 80-column text clearly.

You may be wondering what makes a computer monitor so special that it can do things a 19-inch super-duper remote-control color TV cannot. To help you understand why some monitors cannot display 80-column text very well, it will help to take a brief side

trip inside a monitor and examine a technical specification called bandwidth.

Of Electron Beams and Bandwidth

A video display is often called by another name, CRT, which stands for Cathode Ray Tube. Technically speaking, the CRT refers only to the picture tube inside the monitor or television set. The "cathode ray" part of its name comes from the ray, or precision beam of electrons that emanates from an element, called a cathode, located in the neck of the tube. Magnetic fields around the CRT's neck direct the beam toward the inside surface of the picture screen.

You can think of the screen as a blank painter's canvas, and the electron beam as the artist's brush. In a black-and-white (monochrome) display, the brush paints thin lines (consisting of tiny dots) across the screen, from left to right starting at the top of the screen. Each dot on the screen is called a picture element, or pixel, and it can have many brightness levels depending on the intensity of the beam on that spot. As each succeeding line down the canvas is painted, a recognizable picture appears.

Now imagine that the brush paints each line so fast that it fills an entire canvas with over 300 lines every 1/30 second. That's how frequently a standard television has its picture painted on the tube to present the illusion of motion. Each completely painted screen is called a frame.

The canvas of a CRT is actually a phosphor-coated glass surface. At the instant the high-voltage electron beam strikes the phosphor, that very spot glows. The

greater the intensity of the beam, the brighter the phosphor spot glows.

A color monitor works on the same principles, but with the added complexity of usually having three separate electron beam guns in the neck of the CRT. Each gun is focused precisely on separate red, green, and blue phosphor dots clustered together at the location of each picture element. Since each color's intensity is controlled individually, the colors at a pixel can be mixed to produce the illusion of a broad range of colors when viewed from a comfortable distance. We'll have more to say about color monitors later.

Bandwidth is a difficult concept for a nontechnical person to grasp because it takes into account a couple of esoteric but essential things going on inside a video monitor. In general terms, a bandwidth specification can be likened to the measure of an opening in a tube through which you are trying to push something, like a liquid. The bigger the tube, the more liquid will pass through that tube over a given amount of time.

To demonstrate how this affects a computer's video display, let's use some paint and a spray gun to create the figure-eight character shown in Figure 3-2. The can of paint, acting as our computer, has a regulator on it that sends a measured burst of paint corresponding to each dot. The spray gun represents the CRT's electron gun, and the wall is our phosphor screen. The hose linking the paint can and gun represents the video monitor's circuitry that determines how many bits of computer information can pass to the gun at any given moment.

First we've got to get our spray can moving up to

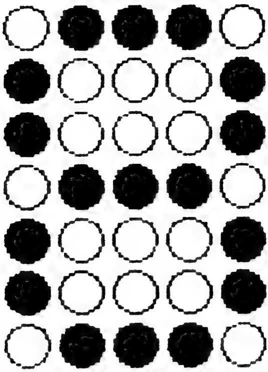


Figure 3-2. **A typical dot pattern of the numeral 8.**

speed, steadily sweeping across the wall from left to right, row by row. Then, if we have a wide enough hose, sufficient paint can pass quickly through the gun with every burst to produce a perfect dot on the wall. But if the hose is too narrow, it takes longer for all the paint to get to the nozzle, and thus for the burst of paint to spew out of the gun. Instead of producing a crisp dot, the burst of paint reaches the wall looking more like an oval, elongated as the gun sweeps to the right. By the time the entire character is painted, it looks smudged.

The *band* part of the term bandwidth comes from the convention of measuring the speed of bits flowing through circuits as if it were a frequency, just like a radio band (e.g., shortwave bands). Bandwidth is measured in megahertz (*hertz* is the recently renamed "cycles per second" measure, named after electricity pioneer Heinrich Hertz), or millions of information

bits per second. TV sets, which have bandwidths in the 3.5 to 4.5 mhz range, don't have enough bandwidth (in our analogy, we would say a wide enough hose) to show 80 characters across the screen with any legibility. Color monitors with bandwidths in the 10 to 12 mhz range (between TV sets and high-quality monitors) produce fuzzy-looking letters blending into one another. High-quality color monitors with bandwidths of 14 mhz and higher produce excellent color and very readable text. A fourth category, monochrome computer monitors, gives the most legible text displays due to high bandwidth ratings and the fact that each picture element consists of a single dot instead of a cluster of colored dots.

In the final analysis, you should choose the type of monitor that offers the best bandwidth for the type of work you will be doing with the computer. Here are your choices for the *II*c.

TV Sets

We've already seen that a standard television set is the least efficient and lowest-quality video display device you can hook up to your *II*c, because it has a low bandwidth compared with the other monitors we'll be looking at next. The bandwidth is insufficient to display legible 80-column text and accurate colors in the higher color graphics resolutions. Attempting to use a TV set with some software will result in a very disappointing display.

Another reason a TV is your worst choice is that the computer video is going through a number of extra conversions on its way to the picture tube. An external RF modulator changes the computer output to a

broadcast frequency on channels 3 or 4. Inside the TV set, that signal is then deconverted back into a video signal like the one initiated by the computer. Every time you convert a video signal, you lose some of the original quality.

Despite the TV's low picture quality, Apple includes everything you need to hook up the *II*c to a television set so you can start using the *II*c within minutes after unpacking it. An RF modulator box snaps onto the back of the *II*c console where the TV set connector is located. The supplied video cable plugs into the RF modulator and extends to the rear of the TV set, where it plugs into the supplied antenna switchbox. You need a screwdriver to attach the TV antenna lead-in to the switchbox and the switchbox to the antenna terminals on the TV, but it is a simple procedure. Later, if you add another monitor, you use the same video cable, as we'll see in the next couple of sections.

An RF modulator set to channel 3 or 4 presents some difficulty in high TV station density areas like New York City, where both channels are used by TV stations. There is the possibility that you won't get a particularly clear computer picture because strong TV station signals on those channels will still be reaching the inner circuits of the set. In such situations you would be better off choosing a different kind of monitor.

The other potential problem of using a television set for a computer display is that it may create a conflict in the home when someone wants to watch *60 Minutes* while someone else wants to print out a term paper. Except for game playing and educational activities, the

Apple *II*c will probably be better installed in a den, bedroom, or otherwise separate area with its own video monitor. Fortunately, the *II*c can be easily moved from the color TV set in the family room to another video monitor elsewhere, depending on the computing needs of the moment.

Color Video Monitors

Up the price scale from television sets is a class of display called the composite video monitor. Its name comes from the type of video signal that it accepts: a composite of all the necessary video information in one signal line. In its most basic form, a composite monitor is nothing more than a television set without a channel tuner. Then why is it more expensive?

Composite monitors are usually manufactured with a higher grade of parts than standard television sets. For example, you may have seen a new kind of home television system called component TV. The video part of the system is a composite monitor, while the channel tuner is usually a separate box. By putting higher-quality electronic parts into the monitor, the display often has a wider bandwidth, resulting in greater clarity of color information, although not enough for 80-column text display. Additionally, the video signal from the computer goes through two fewer conversions. Thus, for color games and educational software, a composite monitor will offer better video quality than an RF-modulated television set.

The bad news is that the improved bandwidth of a composite monitor is still not quite good enough to handle the 80-character-wide text mode of the *II*c. To have both color graphics and 80-column text on the

same display, you need to go to the most expensive class of video display, a high bandwidth direct drive monitor, also called an RGB (red-green-blue) monitor. By late summer 1984, Apple will have a special "docking module" that will let you connect an RGB monitor to the video connector on the *//c* rear panel. While an RGB monitor will produce excellent color graphics displays, it still doesn't do as good a job with text as a monochrome display designed for that task.

The Apple *//c* Monitor

If your work with the *//c* involves just about entirely text work—word processing, spreadsheets, electronic filing, telecommunications—then the newly designed Apple *//c* monitor should be a top-priority addition to your system. The Monitor *//c* is a 9-inch green screen display that plugs directly into the video monitor connector on the rear of the *//c* console.

Styled to match the *//c*, the Monitor *//c* also has an optional cantilevered monitor stand that holds the screen over the console, while keeping plenty of ventilation space above the air holes on the top of the *//c* console. The stand allows some adjustment in the tilt angle of the display to help reduce glare from overhead lighting. The monitor, stand, and computer make quite an attractive-looking workstation (see Figure 3-3). And the green screen (which is also capable of displaying graphics in monochrome) is much easier on the eyes over long work sessions than white characters displayed on a black-and-white monitor.

Flat Panel Display

Expected from Apple by the end of 1984 is a flat panel display that not only matches the *//c* in styling, but

***Figure 3-3.* The matching monochrome monitor provides an attractive workstation.**

makes the computer much more transportable than any Apple *//* has ever been. The display is only about one inch thick, and rests atop the *//*c console (when in use, it loops into the ventilation holes to hold it secure) and gets all its power and signals from a single cable plugged into the same connector used for a television set.

This panel display is at the leading edge of liquid crystal display (LCD) technology because it will be one of the first commercial products to show 80 columns by 24 lines of text on a single panel, replicat-

Figure 3-4. **The liquid crystal display (LCD) flat panel display option for the Apple //c (available late 1984) gives you more portability.**

ing the entire field of a video display. Previously, a 16-line display was about the largest on a commercial product. As soon as someone produces a battery pack for the computer, you will have a truly portable Apple *//* computer.

PRINTERS

Just about every businesslike application requires a printout of some kind. This is especially true with word processing. But even if you do home budgeting on a spreadsheet kind of program you'll probably want a printout of the monthly budget for reference.

Apple offers a choice of two printers—the Imagewriter and the Scribe—and one plotter for the *IIc*. If you can define what you'll be doing with your *IIc*, your printer selection won't be too difficult.

Both the Imagewriter and the Scribe are dot-matrix printers, while the Scribe is also a thermal transfer printer. Let's see what these terms mean to you.

Dot-Matrix Printers—The Imagewriter

Dot-matrix printers get their name because the printed characters are fashioned out of a series of dots. *Matrix* means that the space on the page alloted for a character is actually a rectangular grid of dots just like a video display character (see Figure 3-2). At the intersection of various points in the grid, a dot is printed. In the right order, the printed dots form a recognizable character.

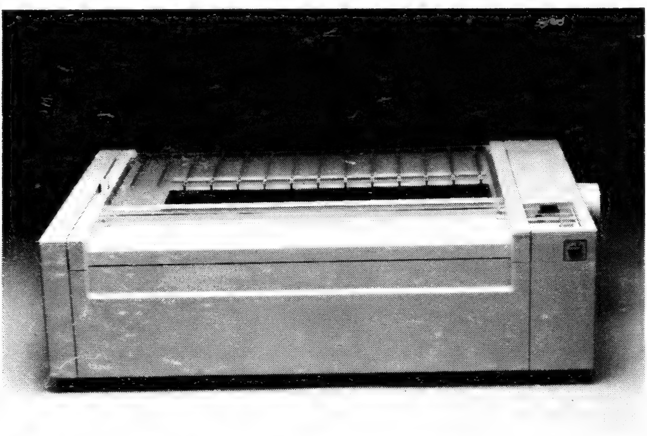


Figure 3-5. **Apple Imagewriter dot matrix printer.**

The mechanics behind a dot-matrix printer like the Imagewriter are quite fascinating because everything moves so incredibly quickly. A typical dot-matrix print head consists of a single column of 8 or 9 pins, each pin with its own plunger behind it. One of the printer's motors draws the print head rapidly across the width of a sheet of paper. As it moves, various plungers thrust pins against an inked ribbon, transferring a dot onto the paper. Let's say we want to print a line of 80 capital letter Cs. In a 7 × 9 matrix (as on the Apple Imagewriter), each letter C is composed of 13 individual dots. At the rated speed of 120 characters per second (cps), the printer is making 1560 dot impressions per second. It's hard to comprehend that anything mechanical can operate with reliability at such speeds, but it does.

Legibility of dot-matrix printer output varies from model to model, and is largely a function of the density of the dots used to form each character on the page. Just like a mosaic mural, the smaller and more closely compacted the component parts of an image are, the more continuous the lines and detail of the image appear.

Dot-matrix printers are also capable of reproducing computer graphics. If you recall our discussion about video displays, you'll remember that what you see on the screen is really a collection of dot images. In printing screen graphics, the computer translates those dots into printed dots. If you choose a printer other than one of the models offered by Apple, remember that not all matrix printers are capable of graphics, however, so you must choose your printer carefully if you intend to print graphics. Common phrases de-

scribing graphics compatibility are *dot addressable graphics* and *all points addressable*. They both mean that the computer can tell the printer to print a single dot in location X if needed. Printers without this feature respond by printing only the characters stored in them at the factory (their built-in character sets).

The Apple Imagewriter is one of the most versatile dot-matrix printers you can add to the *II*c. This is the same printer that prints out superb graphics and text when used in conjunction with the Apple Macintosh. Built into the printer's character set are the 96 standard ASCII characters (capital and lowercase letters, numerals, and punctuation), plus 25 international language characters. With some programming expertise, you can also create your own typeface, if you like.

Under the control of some software programs, the Imagewriter is capable of changing among pica pitch (10 characters per inch, or cpi), elite (12 cpi), and compressed (17 cpi) print in text, as well as several graphics modes. Not all software gives you this flexibility, however. Since the Imagewriter is a relatively new printer in the Apple line, it may take a year or so for producers of text-oriented programs to take full advantage of the Imagewriter's range of powers. But if one of the reasons you buy an Apple *II*c is to print out graphics, then the Imagewriter is the most logical choice and is immediately usable with graphics programs like MousePaint (see Chapter 9).

Thermal Transfer Printers—The Scribe

Apple's new printer, The Scribe, is a remarkable advance in low-cost printer technology for a couple of

reasons. First of all, it uses a thermal transfer process, which differs from a popular process, called thermal printing, in that the Scribe prints on standard paper, instead of specially treated paper. Therefore, you can print on letterhead or a preprinted form, if desired. Thermal transfer printing creates dot-generated characters similar to standard dot-matrix printers, except instead of using tiny hammers to strike an inked ribbon, the transfer print head uses heated hammers which cause a chemical transfer from the special ribbon to the paper. Like ordinary thermal printing, the thermal transfer process on The Scribe is so quiet that you can't believe anything is printing until you see the lines of text start scrolling up the page as it advances line after line. The color of the printing is dependent not on the paper (as with regular thermal printers) but on the special ribbon. Apple offers an optional four-color ribbon for The Scribe, making the printer perhaps the least expensive color printer available for a personal computer.

Another striking feature of The Scribe is that it is one of few printers in its price class to offer both draft- and correspondence-quality printing. Draft quality is comparable to standard dot-matrix printing, in which you can usually distinguish individual dots comprising the letters. Correspondence quality condenses the dot spacing slightly to produce a character that looks closer to a fully formed typewriter character. A trade-off for the higher-quality print is a slower printing speed. In draft quality, The Scribe is rated at 80 characters per second (cps), while in correspondence quality, it is rated at a still respectable 60 cps.

Apple Color Plotter

It's not likely that many users of the *//c* will have need for a color plotter, but if the kind of work you intend to do with the *//c* is graphics oriented, then Apple is ready for you. A plotter is a highly mechanical device, much like a robot arm that actually draws on paper with a pen.

Under the control of software specially designed to take advantage of its unique characteristics, the Apple Color Plotter moves a pen back and forth along a straight line, while another motor slides the paper back and forth in a direction at right angles to the movement of the pen. The result is the plotting of smooth, precise

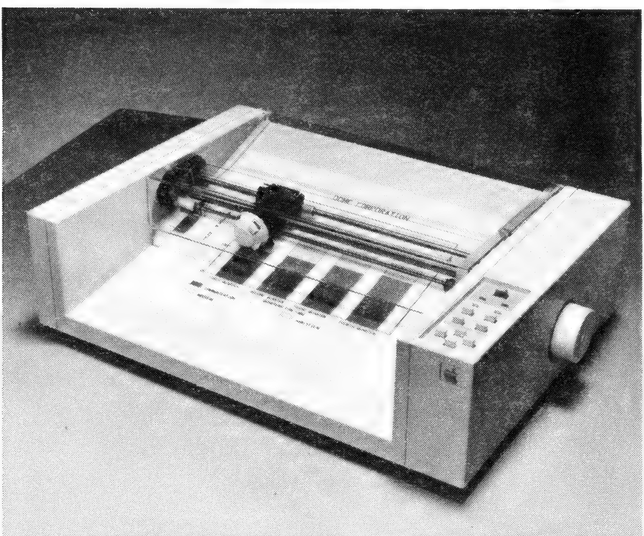


***Figure 3-6*. Apple Color Plotter.**

lines that usually look better than those hand-drawn by an artist. The Apple Color Plotter can also stop during the plotting of a graph or chart, allowing you to change the color of the pen the mechanical arm is drawing with.

The Apple Color Plotter is an expensive, special-purpose accessory for the *//c*. But some applications make good use of it. It is particularly well suited to those people who need to translate scientific and business data into charts and colorful graphs for presentations. The Plotter can even draw those charts and graphs on transparency film for display on overhead projectors. A school might also want to have one plotter that many students can share so they can plot out their graphics designs using the Logo programming language.

Connecting Apple Printers

All three Apple-brand printers mentioned above plug into the printer port on the rear panel of the *//c* (each also needs a 110-volt AC power outlet). None of the printers comes with the cable to attach it to the computer, but each printer has a special accessory kit available that includes the proper cable designed for the *//c*.

Using Non-Apple Printers

You're not limited to using Apple-brand printers with your *//c*. But you must make sure that whatever printer you select is compatible with the computer and software you run, and also that the proper cabling is available.

The first thing to be on the lookout for is that the printer you select should have a serial interface, not a

parallel Centronics interface, to be compatible with your *//*c hardware. The terms serial and parallel refer to the manner in which computer data is sent to the printer.

In a serial printer setup, there is one data line in the cable running from computer to printer, plus a couple of other control lines. To assure compatibility among all serially connected devices (printers, modems, etc.), the makeup of signals sent over each line has been standardized by an industry group. The number assigned to this Recommended Standard is 232. Therefore, whenever you see reference to RS-232 cables and connectors, simply substitute "serial."

If you recall our discussion about bits in Chapter 2, you'll remember that it takes 8 bits to make up a distinct character code. In a serial transmission, the bits are sent like 8-car trains down that one data track. A parallel printer, on the other hand, gets all the bits for a single character in one burst, with each bit going down a separate data line. But since the Apple *//*c doesn't have a parallel printer port built in, you can't run a parallel printer directly from the computer. Eventually, you will be able to buy a serial-to-parallel convertor that plugs into the *//*c serial printer port.

As you might imagine, the parallel transmission of data at a given speed is faster than serial transmission. In the time it takes for one bit to pass from computer to printer along a serial line, a complete character has reached the parallel printer. But, as it turns out, personal computer printers do not operate fast enough to be affected by the speed differences. In fact, they run so much more slowly than the speed at which the computer normally sends data that during the printing

of a document a printer often tells the computer through one of the control lines to hold up sending any more data until the printer is ready again.

You must also be cognizant of potential problems with printer-software incompatibility. Special functions like underlining, compressed print, and boldfacing require special printer codes sent by the particular software program you're running. Not all programs send the right control signals needed by all printers. It's possible that you can spend a virtual fortune on a fancy printer only to discover that the programs you have won't let you take advantage of any of the printer's fancy features.

Programs that rely heavily on printers—word-processing and graphics programs primarily—frequently contain initializing routines which you have to complete the first time you operate the program. When you initialize a program, you usually select the printer you're using from a listing on the screen.

If your computing needs are in the word-processing or graphics areas, you would be wise to first locate the software you want to use and then see with which printers the programs are compatible. Then buy the printer. You'll save yourself a lot of headaches following this procedure—a printer mismatch can be a costly nightmare.

Another thing to watch out for when selecting a non-Apple printer is that the printer must have a cable available specifically configured for the Apple //c. The cable is almost always an extra-cost option (ranging from $30 to $60), and is the critical link between

computer and printer. If you're handy with a soldering iron, you might consider wiring up your own, but it takes a steady hand and a comfortable knowledge about the different signal lines running from the *II*c serial port and the printer's connector.

Formed-Character Printers

If you're serious about word processing on the *II*c, you may want to invest in a formed-character printer, even though such printers may cost more than the computer. A formed-character printer gets its name from the mechanics behind the composition of each printed character, in which an embossed printing element is thrust against an inked ribbon, forming a complete character in one hit. A typewriter does just this, of course, but computer printers are designed to operate at faster speeds and longer stretches than most typewriters can withstand.

Two popular types of formed-character printers are the printwheel and thimble type. In the former, all printable characters are molded onto the ends of spokes connected to a printwheel hub. A print thimble operates on the same principle, except the spokes are bent at a right angle to the hub, forming a broad thimble shape. As the print head mechanism moves across a line, the wheel or thimble spins amazingly fast to position the proper character in front of the single hammer for the striking blow.

Formed-character printers operate at slower speeds than do dot-matrix impact printers. Low-cost models for under $700 print as slowly as 10 characters per second. Faster models, from about 35 to 55 characters per second, can cost from $1500 to $3000.

If the *II*c is going to be used for typing schoolwork like term papers and reports, then a formed-character printer or a dot-matrix printer capable of letter-quality appearance is recommended. Teachers and editors are generally not fond of reading low-resolution dot-matrix printed documents: they tax the eyes after a while. The good thing about making a heavy investment in a printer even with the *II*c is that you will be able to use that printer with a more sophisticated computer if you should upgrade some years later.

MODEMS

The term modem stands for MOdulator-DEModulator, and it is a piece of equipment you must have to connect your computer for communications through a telephone line. In our earlier discussion about the workings of computer circuits, I mentioned that all communication between circuits was in streams of on and off pulses. In order to send these pulses across the telephone line, they must be converted in some way to audible signals. At the transmitting end, a modem performs the conversion from electrical pulses to audio signals; at the receiving end, a modem performs the opposite conversion. If you were to eavesdrop on the telephone line connecting two computers, you'd hear streams of warbling tones. In most connections, the two computers are transmitting and receiving simultaneously in an arrangement called full duplex—just like the way some people use the phone, when both parties talk and (presumably) listen at the same time (unlike a CB radio where only one party can talk at a time).

Connecting the *II*c to the phone line via a modem is a simple task. Think of the modem as an intermediary

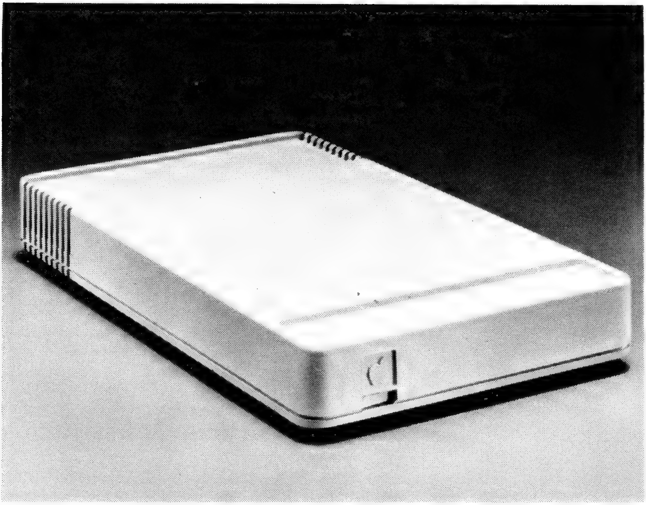


Figure 3-7. **Apple offers single speed (300 baud) and dual speed (1200/300) baud modems for the //c.**

box in the middle of the cable running from the computer's modem rear panel connection to the phone jack in the wall—just as the Floor Mount transformer is located in the middle of the power cable. You can plug a telephone into the modem, as well, if you want to use the telephone wall outlet for both computer and human communications at different times.

Apple offers two modems for the //c—the same modems used by all Apple computers. What distinguishes them (other than price) is an important specification that may sound like gibberish to a computer novice: the baud rates. Baud (pronounced *bawd*) is short for Baudot *(bawd-doe')*, the name of a communications pioneer in the days long before computers. A baud is a

unit of measure of the speed at which bits of information are being passed along a communications link (sometimes radio, as well as wire). One modem model is rated at 300 baud, which means that 300 bits of data are sent in one second. In most common computer communications, you need eleven bits of data to send the equivalent of one character. Seven of the bits are devoted to the code for the specific character. The rest are extra signals the receiving station needs to help it know when one character ends and a new character begins. Thus, a speed of 300 baud translates into a transmission rate of roughly 30 characters per second. At this rate, about 270 words per minute, you can read the stream of incoming data on the screen without much difficulty.

It may be desirable sometimes, however, to communicate at 1200 baud, the top speed of the second Apple modem. While most commercial services like CompuServe and The Source have 1200-baud service as an option, they charge considerably more for service at this transmission speed. You should study their charges carefully, because there may be an advantage to running 1200 baud if the cost for this four-times-regular speed is only double the 300-baud connect rate.

But the effective use of 1200-baud service also depends a lot on the type of communications you're doing. Since text will display far faster than you can read it at this rate, high-speed data transfer is most cost-effective when you are sending or receiving large documents that don't need to be read as they are winging their way along the phone line. On the other hand, if the communications require a lot of inter-

change between the two parties (like having to respond to a lot of menu choices along the way), you may be better off at the lower-rate 300-baud service. Otherwise, you'll be wasting the more expensive connect time just to read the menus. The best way to cover yourself for most applications is to have the more expensive, dual-speed Apple modem, which is capable of both 1200- and 300-baud data transfers.

Of course it takes more than just a modem and a computer to start communicating. You must also have software that converts the computer and modem into a communicating terminal. With a communications program, you can have the computer automatically dial the necessary phone number to get onto the electronic data-base service and send the proper log-on identification codes and passwords. The program maintains a dialing directory to which you can add new services or local computer bulletin board phone numbers as you go along. You can also set up the computer to answer the phone (after a certain number of rings) so it can communicate with other computer users who know your computer is "on-line," even if you're not there.

Most new computer users who have never been exposed to the wonders of telecommunications pass over this application, thinking that it's not for them. But the variety of services available, the ease with which the *//c* makes computer communications possible, and the people you can meet with similar interests on The Source and CompuServe will amaze you. I recommend that you add a modem and communications software early in your *//c* learning process. It may well open up a new gateway to home computing you never even imagined.

EXTERNAL DISK DRIVE AND FURTHER EXPANSION

If your applications require a lot of swapping of program and storage disks, then you may want to add a second disk drive to streamline your computer work. The external //c disk drive is the same drive mechanism as inside the main console, but housed in a separate, matching low-profile box. All control signals and electrical power for the external drive come through a single cable, which attaches to the largest D-type connector on the //c rear panel.

The external disk drive connector, by the way, may be used in the future by Apple and outside developers for more substantial expansion possibilities. Someone, for

***Figure 3-8*. External disk drive for the //c.**

example, could come up with an outside expansion chassis to allow the attachment of accessories not easily adapted to the serial ports already in the *//*c. The connector will be a gateway for inventive hardware enthusiasts, to be sure.

APPLEMOUSE POINTING DEVICE

Several months before the introduction of the Apple *//*c, Apple announced the availability of a mouse pointing device and mouse adapter for the Apple *//*e. A mouse is a handheld device, about the size of a deck of cards, with a small roller underneath that measures the direction and distance you slide the device along a desktop surface with one hand. The measurement is converted inside the computer to signals that control the location of a pointer on the video screen. For example, if several program commands are displayed in a line, you can use the mouse to move an on-screen pointer to the particular command you wish to give. Typically, to visually confirm that you are pointing to the desired command, it will be displayed on the screen in some highlighted form (usually in inverse video). On the mouse is also a button. To proceed with the highlighted command, you press the button—a step Apple calls *clicking*.

A mouse is one alternative to using step-by-step cursor control keys, which work only in one of four directions at a time (up, down, left, right). With a mouse, however, you are free to move the pointer around the screen, cutting any diagonal you need in order to get to the desired position quickly. It is the closest you can get to having a touch-panel screen, since your hand is actually controlling an electronic

Figure 3-9. **The AppleMouse comes with graphics software.**

"finger" on the screen. Yet you don't have to lift your arm from the desktop to make it work.

Mouse pointing devices are a source of controversy, since some dyed-in-the-wool keyboard enthusiasts fail to see how a separate desktop gadget, which forces you to pull one hand away from the keyboard, can possibly be productive. But properly designed mouse-driven software reduces the number of keystrokes

needed to accomplish on-screen tasks, such as selecting options from menus. And if the program design includes many mouse functions in sequence, you don't get the feeling that you are constantly shifting from keyboard to mouse and back to keyboard.

At first, using a mouse may seem awkward. But it doesn't take longer than about five minutes to get the feel of how the pointer responds to your movement. In short order, you will be able to concentrate fully on the screen action and forget that the mouse is involved at all. In graphics-oriented programs, the mouse becomes an extension of creative talents you never knew you had.

The AppleMouse for the *//c* comes packaged with a marvelous graphics program, called MousePaint. This program is fashioned after the revolutionary Macintosh program, MacPaint. You will see how MousePaint works later, in Part II. The mouse attaches simply to the game controller/mouse connector on the rear panel of the *//c*. No power cord or batteries are needed.

GAME CONTROLLERS

Available for the *//c* are a cosmetically redesigned series of paddle controllers and game joysticks. The new design simply reflects the new direction in styling and coloration for Apple *//* products, more closely matching that of the *//c*. Even the connecting cables are color coordinated.

Expected at a later date from a third-party manufacturer is an adapter that will allow two joysticks to be

connected to the computer at one time. Currently, only paddle controllers can be attached as a pair (they are constructed as one pair, built into a single connector).

CARRYING CASE

The transportability of the *II*c system is emphasized by the fact that the computer, the Floor Mount transformer, cables, disks, manuals, mouse, RF modulator, and flat panel display (when it becomes available) can all be packed into a *II*c system carrying case that is scarcely larger than a briefcase. It is constructed of a durable cordura cloth material, and adds very little weight to the entire package.

The interior of the case is compartmentalized for the main console, floor mount, and cables. It is hoped that a battery pack can eventually be built that fits in the same volume as the Floor Mount transformer, thus giving you a completely portable computer in a case. The mouse and flat panel display fit in outside pockets.

Now that you know what the *II*c is, and what things can be added to it, it's time to tackle the subjects of how the *II*c compares with a number of other computers on the market, and whether the *II*c is the computer you should buy.

CHAPTER 4

Is the Apple *II*c for You?

By now, you should have a pretty good understanding of not only what the specifications for the Apple *II*c are, but also what they mean to a computer user. Armed with this knowledge, you can begin to sort out the various personal computers currently on the market and see whether the *II*c is right for you. Before we tackle individual competing models, however, it is important to understand the fundamental positioning of the kinds of personal computers you see for sale in stores ranging from the national discount chain to the neighborhood business computer specialist.

THREE PERSONAL COMPUTER TYPES

Over the past few years, personal computers have become stratified into three definable layers, each appealing to a different customer. At one end are those computers intended primarily for home use—models like the Atari 600XL, Coleco Adam, and Commodore VIC-20 and 64. These computers are sold in discount,

toy, and even drug stores at rock-bottom discount prices. The prices, of course, are usually for the basic console only (a notable exception is the Coleco Adam, which includes a letter-quality printer and a couple of pieces of software). Since the low prices for these computers attract consumers who are upgrading from home video games, software developers for these computers devote most of their energies to entertainment software. Educational and productivity software is severely lacking, and what is out there for these computers is by and large inadequate for the productivity expectations of the people who buy the computers.

At the opposite end of the price and performance spectrum are professional and small-business personal computers. These computers—the IBM PC, PC-compatibles, Apple Macintosh, Kaypro portables, and many others—are designed to serve business people and their companies. Many come equipped with several hundred kilobytes of RAM and can be configured with a rigid disk drive capable of storing in excess of 10 million bytes (1000K) of programs and information. The kinds of programs these computers run are the heavy-duty productivity packages that professionals demand. The computers keep paperwork in order in small medical and legal offices. They give high-volume professional writers the ability to manage large or multiple documents. Financial experts can work with electronic spreadsheets whose ledger-sheet equivalent would be large enough to paper a wall. Business graphics can be at a tremendously high resolution, as on the Macintosh. The latest software developments for these machines allow several different applications to be running on the screen at one time.

Between these two kinds of computers is a gap, not only in price but in capability. Demand for computers in this gap comes from those who need a computer to perform businesslike tasks on a modest volume with a modest budget. The results need to be professional-looking, but the amount of information being managed doesn't require 10 million bytes of hard disk storage or the simultaneous display of three documents. At the same time, the computer may often find itself being used at home, either because that's where the businesslike work is done, or because the computer is shared with children who use it for educational and entertainment purposes. Filling this gap today are really only three computers, the IBM PCjr, the Apple *//e*, and the new *//c*.

The first decision you must make, therefore, is into which of these three categories your computing needs fall. Let's see what computers at the low and high ends offer in comparison with the *//c*.

APPLE *//c* vs. THE LOW END

Comparing the *//c* to any of the low-end home computers is almost pointless. The vast majority of programs for low-end computers continues to be of the entertainment variety. Memory capacities of these machines are limited to 64K (with the exception of the Adam, which has only slightly more, distributed among its components). For productivity software, you'll have to look hard, and still probably not find the program that does the kind of problem solving you'll need in the long run. Most important, none of these home computers display 80 columns across the video screen. Most have only 40-column displays, making it

cumbersome to work with productivity programs that display a lot of text on the screen. You spend more time scrolling through and across pages to look for the information you need than actually working with the information. Software demonstrations with all kinds of fancy movement and graphics on the screen don't mean much when the program fails to be truly productive or sophisticated enough to handle the amount of information you're typically working with.

The pricing of many so-called budget systems can be deceiving, by the way. Computers in the under-$200 class need a disk drive in the $250 price class to be anywhere near productive. Attaching other peripherals often entails the purchase of further adapters. By the time you've assembled a complete system, you've spent many times the $200 advertised price. And you're still left with an underpowered, 40-column computer that has mostly game software available for it. These are the computers that most often get put away in the closet not long after they are purchased.

APPLE *//c* vs. THE HIGH END

We've already noted that the high-end personal computers are more capable in high-volume productivity situations than the *//c*. If you know that your applications demand instantaneous recall from a large base of information (e.g., a thousand inventory entries, several hundred client records, etc.), then computers capable of hard disk storage like the IBM PC XT, Apple *///*, and Apple Lisa are more suitable than the *//c*. Or if you need business solutions that are best handled by software available for the IBM PC and computers compatible with the PC that take advantage

of expansive memory, then these are the kinds of computers you should be investigating, even though they will be considerably more expensive than a *//*c system. Similarly, if high-resolution graphics and interchangeability of information from one program to another are useful in your work, then give the Macintosh or Lisa careful consideration.

APPLE *//c* vs. IBM PCjr

Industry watchers like to pit Apple and IBM against each other, almost for a sport. The two companies are, of course, the two biggest players in the personal-computing game, so I suppose this kind of thing is bound to become popular. With the arrival of the *//*c, the competition is intensifying.

Like the *//*c, the PCjr is intended as a meaningful and useful home computer that can be called into service for light-duty business work. Both computers are in a similar price range, so the choice you make should be based on other factors, namely the kinds of things you can do with them and ease of use.

Software Selection

Today, the *//*c is a clear winner when considering the amount and variety of software currently available. Considering that roughly 90 percent of Apple *//* software titles on the shelves today are compatible with the *//*c, it is clear that there is a much wider choice of software for the *//*c than for the PCjr. Until the introduction of the *//*c, the balance of software power was starting to shift toward the PCjr, as software designers gladly worked on programs that took advantage of the PCjr's expanded memory capabilities. Although the *//*e

was expandable to 128K of RAM, software designers preferred to stay within the base unit's 64K, sacrificing program sophistication but thereby making their products usable by every *II*e owner, not just those who added the memory expansion.

But with the arrival of the *II*c, software developers have a renewed interest in creating software for the Apple *II* series. The fact that every *II*c comes equipped with 128K of RAM, 80-column text display, and a mouse adapter allows designers to freely use such programming enhancements as double-high-resolution graphics and Mouse Text graphics characters in their programs. The added memory also leaves plenty of room for more sophisticated productivity programs.

Apple is also engaged in a program to encourage all software developers to adopt a consistent command language for their programs. In other words, in time, more and more programs (particularly productivity programs) will use a command like Open Apple-C (the simultaneous pressing of the Open Apple and C keys) to signal a Copy command. What this will mean is that you won't have to learn a completely new command language every time you buy a new productivity program. In computer jargon, by the way, this approach is called a consistent user interface. Eventually, the *II*c user interface may well be directed toward using a mouse and "pull-down menus," just as the Macintosh uses its mouse to show menu options and make command selections. The MousePaint program (see Chapter 9) is a peek at the future for Apple *II* software.

Assembling a System

When you buy a PCjr, you have a little more work cut out for you than when you buy a *II*c. To the basic PCjr

unit, you must add optional cables for connection to a TV set or monitor, and an RF modulator as well if you use the computer with a TV set. A keyboard cable is recommended, since the wireless keyboard feature has some limitations. Inside the *//*c box, however, is everything you need to get the computer going on your TV set without having to add a single option. If you plan to add a printer to the PCjr, most printers require an additional adapter (called a parallel printer attachment), except for the IBM Compact Printer, which, however, does not print on plain paper. All of Apple's plain paper printers for the *//*c, however, plug directly into the serial port built into the console.

Keyboards

Comparing the keyboards on the PCjr and *//*c must be handled carefully. Keyboard selection is a very personal process. One person's ideal keyboard is another's horror story. The keys on the PCjr lightweight detached keyboard can feel stiff to many fingers. At first glance, it appears that the PCjr keys are set farther apart than on a standard typewriter keyboard, but the actual center-to-center measurement is the same for the IBM PC, Apple *//*c, and PCjr. What the spaces between the keys let IBM software suppliers do, however, is include keyboard overlays with some of their programs. An overlay is a piece of cardboard with cutouts for all the keys. It rests on top of the keyboard surface without obstructing access to the keys. Printed on the overlay are standard key legends and special color-coded legends that link function keys together, so that all the commands are visible from the keyboard. Thus, you don't have to remember a command language or have to refer to a manual when you forget how to perform a rarely used command. This is definitely an advantage to beginners and youngsters.

The trade-off, however, is that the PCjr keyboard is not comfortable for long stretches of typing, as when it is used for writing with a word-processing program. For productivity programs, for which keyboard entry is typically important, the pleasing touch of the typewriter-style //c keyboard is preferred.

Disk Storage

The amount of disk drive storage is a valid point on which to compare the two computers. The PCjr's single disk drive can store up to 360K of information, while the //c's internal drive can hold only up to 140K. This means that on some of the more powerful //c programs, you may have to load in two program disks before you begin using the program, while on the PCjr, one disk can cover it all. The disk storage capacity's greatest impact on your work, however, is when you use disks for storage of information you enter into the computer—word-processing documents, mailing lists, etc. A double-sided PCjr disk can hold more than two and a half times the information of a //c disk before you have to change storage disks. But there's an unusual twist to this story.

In many high-powered productivity programs, the single disk drives of the PCjr and //c may prove cumbersome because you have to swap back and forth between your program disk and storage disk. If this disk swapping should become a nuisance on a //c, you at least have the option of adding an external second disk drive. In contrast, IBM has made no provisions for the addition of an external disk drive. Some third-party suppliers are making them, but they are rather costly, because the computer was not specifically designed to accept a second drive.

This is not to say that the IBM PCjr doesn't have some features that make it an attractive personal computer. For one, the PCjr accepts cartridge software as well as disks. Most of the cartridge software available for the PCjr is in entertainment and education. Many kids from computer families have had experience with cartridges using home video games. Cartridges are easy to load and are durable in the hands of youngsters. Peanut butter and jelly on a cartridge does nowhere near the damage it does to an exposed floppy disk.

Also in the PCjr's favor is the fact that most disk-based software you buy for it will be usable on a more powerful computer in the IBM PC family—the PC, XT, and Portable PC. If your company has installed an IBM XT on your desk at work, and you want to bring work home with you, then there is a pretty good chance that you will be able to run the same (or a similar) program on the PCjr, while you use the same storage disks on both machines. If you prefer the Apple *//c* for home use, however, there are programs available that let you transfer information from your Apple computer to an IBM computer (and vice versa), either over the telephone or through a direct connection. You'll have to make sure, however, that the information files used by programs in both computers are in compatible formats once they are transferred.

Deciding between the *//c* and PCjr may not be a simple process for you. The key features to look at are how important the keyboard, ease of setup, transportability, compatibility with other computers and the diversity and quality of software are to you. Both computers are supported by strong, ongoing companies that are not likely to leave you out in the cold by going

out of business overnight, as some other personal-computer firms have. Outside software development for both machines is continuing at great speed, and programmers are expanding the horizons of each computer, giving you more and more powerful programs to work with all the time.

APPLE *II*c vs. APPLE *II*e

Unlike previous new generations of Apple *II* computers, the *II*c does not replace the *II*e on the shelves of dealers. The *II*e will continue as a current product in the Apple line. At first glance, the two computers may seem to be competing for the same customer. But there are significant differences between the two machines, and each should appeal to a different user.

The major distinction between the *II*e and *II*c is the fact that the *II*e has expansion slots inside the console for adapter boards that control peripherals, while the *II*c has the equivalent of the most common peripheral adapter boards built into the compact console. Each setup has its benefits.

Hobbyists

Computer hobbyists (frequently called hackers) are a dedicated corp of experimenters who like to get their hands "dirty" with circuit boards and designing special adapters for unusual applications. Perhaps they're working on using the computer to act like a piece of sophisticated electronic test equipment. To do this, they need access to the computer's expansion slots. Dozens of special-purpose expansion boards that appeal to these hobbyists are already on the market. By the way, you can usually spot a *II*e hacker because he

or she operates the computer almost all the time with the top cover removed so there is instantaneous access to the expansion boards inside.

Less technical computer owners, on the other hand, will more likely prefer not to go digging inside a computer that costs more than $1000. The connectors on the rear panel of the *//*c (and their corresponding built-in adapters) have been selected for the most common expansion needs, such as a printer, a modem, a joystick, a mouse, a second disk drive, and a video monitor. Is there anything else that you'd want to add to the computer? Designers of certain special-purpose adapter boards and peripherals (like music synthesizers) for the *//*e are already developing versions for the *//*c that plug directly into one of the serial ports, eliminating the need to get inside the machine. In fact, Apple is working toward making the serial ports of all its computers (including the Macintosh) the standard gateway to special-purpose peripherals.

In a School

The *//*e may be preferred by educators over the *//*c in many instances. One reason is that most educational software operates with only 64K of memory. If the schools can purchase a 64K Apple *//*e for considerably less than the standard 128K *//*c, then their budgets will stretch further. Likewise, if the computer in the school system is used for networking (linking the computer to other Apples or to a central computer loaded with instructional material), then the memory capacity and peripheral adapter complement can be kept to a minimum, helping keep costs low. And an Apple in the science classroom may be used a lot like the Apple on the hacker's desk. Designing special measuring equip-

ment cards and using the computer in other experiments may entail having access to expansion cards inside a *II*e.

The *II*c's transportability is both an asset and a liability in a school environment. It's great if the computers need to be moved around from classroom to classroom or from station to station within a classroom when resources are thin. On the other hand, a computer the size of the Apple *II*c might just disappear from its rightful location (it fits comfortably in a backpack). Security-conscious educators might prefer that students bring their own *II*c machines from home in their carrying cases, rather than encourage students to borrow computers overnight.

For Business

In a business environment, the choice between the *II*e and *II*c is largely dependent on the applications intended for the computer. The *II*e holds an edge in some cases because it can be configured to accept high-capacity hard disk storage, while the *II*c cannot at this time. Additionally, having the expansion slots on the *II*e allows the computer to add what is known as a coprocessor board—a circuit board that holds an entirely different microprocessor. Of these, the most popular is one that features a Z-80 CPU chip enabling the Apple *II*e computer to run business programs compatible with a disk operating system called CP/M. With this board and operating system, the Apple *II*e can run popular productivity programs like *WordStar* (word processing) and *dBase II* (data-base management), which run predominantly on computers in the high-end category we discussed earlier. The ability to add CP/M

compatibility and a hard disk allow the *//e* to be upgraded to a more businesslike personal computer than the *//c* can now.

But where the *//c* really shines in business is its transportability. If you need to take your Apple-compatible programs with you on the road or work on them at home, then moving the *//c* around is much easier than trying to wrestle with the bulky *//e* and separate disk drive. Moreover, by the end of 1984, the *//c* will be not only transportable, but truly portable. You'll be able to set up your computer just about anywhere and work with the same productivity and software that you use back in the office.

It is quite likely that the Apple *//c* will be purchased as a second computer for use on the road and at home by those who already use a *//e* in the office. The large degree of software compatibility, especially among productivity programs, makes the *//c* a logical choice. Use the *//e* at the office with the hard disk that stores vast amounts of information. Then copy just what you need for your travels onto floppy disks and use the information with the same program in both locations.

At Home

For most home computer consumers, the Apple *//c* is a much more logical choice than the *//e*. First of all, setting up the *//c* is much simpler, since everything you need to get it running and actually *doing* something is included in the box with the computer. Adding peripherals is as simple as plugging the accessory into its own unique connector on the rear panel—no digging inside the computer and fooling around with plug-in boards.

No small consideration in the home, too, is the fact that the Apple *II*c looks more like something you'd want to have in your home. In contrast, the *II*e is much larger and requires at least one extra box—a disk drive—cabled to it before you have a practical system. You can even stow the *II*c in a drawer if you want to get it out of the way for any reason. The *II*c doesn't dominate a room's decor, but rather looks like a compact, high-technology room accessory.

Extra *II*c Features

The new Apple *II*c has a couple of extra features that make it more attractive. One of the enhancements to the *II*c over the *II*e is related to the use of the mouse as a pointing device. Not only is the mouse adapter built into the *II*c (it's an extra adapter card on the *II*e), but the *II*c also has a couple of dozen extra Mouse Text graphics characters built into its character set (a character set is the list of a computer's letters, numbers, symbols, and graphics shapes). These graphics characters can be used to make the video displays more like that on the Macintosh, albeit with lower graphics resolution. Currently, very little Apple *II*c software takes advantage of Mouse Text characters, but when new programs do, you will have programs on the *II*c that are easier to use and more visually interesting than on the *II*e.

Some programs may also operate a bit faster on the *II*c than on the *II*e. This is due primarily to the slightly enhanced capabilities of the 65C02 microprocessor used in the *II*c over the 6502 in the *II*e. Also, programmers have access to some direct commands with the new chip that combine multiple commands from the old, reducing the execution time of some instructions.

Cost Comparisons

One factor that can't be overlooked for a consumer product is cost. When figuring out the cost of an Apple *//*c against a comparably equipped *//*e, the economy indicator swings decidedly in the *//*c's direction.

To the basic *//*e console, you must add several adapter boards to bring it up to specifications equivalent to the standard *//*c. Even when you compare the attractively priced prepackaged *//*e hardware systems offered by Apple and its dealers, you come out ahead for a system including the *//*c, monochrome monitor, monitor stand, and second disk drive.

On balance, therefore, the Apple *//*c is a more consumer-oriented computer than the *//*e. Its compact design is less obtrusive in the home. It's much easier to set up and move around. Coming as it does with several disks of instruction and introduction, plus everything you need to get it working with your color television, it is really targeted toward the first-time computer user in a home environment, even if the intended applications are productivity related. The *//*c also expands easily, without asking you to open the cabinet.

If you've been casting longing glances toward the Apple II Plus and *//*e over the years, the *//*c is the computer you should buy.

PART II

What an Apple //c Can Do

CHAPTER 5

The Apple *II*c as a Writing Tool

If there is one application that can demonstrate the Apple *II*c's work-saving abilities in both home and work environments, it is as a replacement for the legal pad and typewriter. Writing takes on many guises, of course, and it often involves more than just the physical job of putting thoughts down on paper. It sometimes includes researching and organizing large volumes of information in preparation for the actual writing. Other times, the creative process is performed just once and the bulk of the work is used over and over, with perhaps just a few words changed to fit the situation. Regardless of the writing task to be accomplished, the *II*c not only saves time normally devoted to retyping second and third drafts, but it also encourages all writers to fine-tune their work, since with the computer it is no longer a messy chore to make corrections at the very last minute before the work goes out. And the work that does go out is free of erasures and correction fluid.

SITTING IN FRONT OF A WORD PROCESSOR

If you've never used word processing on a computer before, you're in for a treat. Here's what it's like to write with a typical program.

After you load a word-processing program into the computer (with the //c, it's usually as easy as placing the program disk in the built-in disk drive and turning on the computer), you are often presented with a list of options (called a menu), which probably includes creating a new document or making revisions to one that you wrote earlier and saved on disk. If you choose the menu option to create a new document, the next screen is likely to be predominantly blank. Programs like Bank Street Writer (published by Broderbund Software) display further menus on the screen so you're never at a loss for what to do next. Other programs, like the word-processing portion of Apple-Works (Apple), display only the command you need to get further help on the screen. Issuing commands is not as difficult as it may sound. Commands are given to many //c programs by pressing the Control or Open Apple command key together with another key. Because you're pressing one of the command keys, the program knows not to display either character on the screen, but rather to act on a command associated with that key sequence.

Once you're all set in the mode that lets you type your document, you do simply that, just as if you were using a typewriter—almost. Corresponding to the point on the typewriter platen at which the hammers strike the ribbon and paper is a blinking marker on the screen called a cursor. Wherever on the screen the cursor is flashing, that's where the next character you

type will appear, regardless of whether it is at the end of your draft or in the middle of a paragraph. So, instead of rolling the paper up and down or spacing and backspacing the carriage to position the paper in front of the typing element, you take control of the cursor with the help of cursor movement keys (up, down, left, right), and step it over to the precise location you desire. Every character you type appears on the screen and is temporarily stored in memory (until you clear it or save it on disk). Typically, you can backspace over errors and type the correct letters you meant in the first place.

There's no bell on the word processor to ring as a reminder that the line of text is getting full. Why should it alert you to perform the manual task of bringing the cursor back to the left margin of the next line when the computer can do it for you? The computer is smart enough to know when a word you're typing at the end of a line won't fit within the right margin. When it detects such a word, the program automatically erases from the screen the characters you started typing and moves them to the left margin of the next line. You type away, uninterrupted by end-of-line signals. This feature, by the way, is called word wrap, and there's hardly a word-processing program that doesn't feature it as standard equipment.

When you reach the end of a paragraph, it's time to signal the computer to go to a new line to start a new paragraph. To do this, you press the RETURN key, which is located in the same place as the automatic carriage return on electric typewriters. You press the RETURN key again to get a blank line between paragraphs.

With the *II*c's 24-line display, you can't see a complete page of a document. The screen behaves more like a window onto a continuous document that scrolls up and down behind it. Even though part of the document isn't showing on the screen, it's still safe in the computer's memory. If you try to move the cursor "off the screen" outside of the window, the text simply scrolls up to meet the cursor. There are also shortcuts for jumping up or down through long documents.

Word-processing programs do their best to duplicate some of the features you're accustomed to on a typewriter. You can set left and right margins, for instance. You can also set tabs along a line to help you line up your indentations. But a word-processing computer also adds a number of conveniences you probably wished you'd had all along with a typewriter.

For example, most word-processing programs display a count of the number of lines you've typed. Therefore, if you know you have a table of figures you want to include in a document, and you don't want the table to be spread across two pages, you can look at the line counter and determine if there's going to be enough room. If not, start the table on the next page.

Your page layouts are also under your complete control in a way that is a lot easier than on a typewriter. Centering a heading, for example, doesn't take counting the letters anymore. Simply command the computer to center what you're about to type, and it does, perfectly. The computer can also right justify your documents—inserting an extra space between words here and there to make sure the right margin of your document lines up as straight as the left margin. And,

when it comes to double or triple spacing between lines, the word processor usually takes care of that for you, as you command.

Perhaps the most beneficial aspect of a word-processing computer like the *//c* is that even after you've typed in the document, you can play around with the page layout and overall appearance of the document on the screen. You don't have to try to guess how the page would look if you changed a particular parameter. Make the change, and then print out a copy as often as needed until it looks the way you want it to look.

The same goes for making changes to the text. Print out a copy of the rough draft. Note any corrections in grammar, spelling, and organization on the printout; then go to the computer and type in the changes. Moving sentences and paragraphs around on the screen is far easier than trying to cut and paste together sections of a document already typed (you'll see an example of this later in the chapter).

Because you can change a document at the very last minute before it goes out, and because even a relatively slow printer delivers finished documents at many hundreds of words per minute (few professional typists can claim even 100 words per minute), you are more likely to bother making a tiny correction to a letter or report that has to be ready by the last mail pickup. Your output will be more professional-looking and probably better in content, as well, since you won't be reluctant to make improvements.

While it's true that a word processor is the cornerstone of a writer's computer software, there are other

very useful programs that make the job of writing many kinds of documents faster and more professional. To help you understand the various ways the *II*c aids writers with different kinds of documents and software, let's trace the creation, refinement, and final production of three types of documents: a simple letter, a history essay, and a series of personalized form letters.

THE LETTER

Producing one-page letters on a *II*c doesn't require a particularly sophisticated word-processing program. A word-processing program like Bank Street Writer (Broderbund), although capable of producing much longer documents, is a program you can use for occasional correspondence with great ease, since all commands and options are displayed on-screen along with the typed text.

The Bank Street Writer screen presents a text window, where your text appears. You type the date, addressee, salutation, body, and closing just as you would on a typewriter. Moving any section of text is a simple procedure. You select the Move option from the Edit Menu (displayed when you press the Escape key) by moving a highlighted cursor, which is activated by pressing the left or right arrow keys, to the command word, Move, and pressing Return. A screen prompt asks you to move the cursor to the first character of the text to be moved and to press Return. Then you're told to move the cursor to the end of the text block. As you do, all selected text is highlighted, showing you precisely the amount of text about to be moved. Finally, you move the text cursor to the point of insertion for the block and press Return. If you

don't like the move, you can issue the Move Back command to start over.

The appearance of a letter is important if you are trying to impress the recipient with a professional presentation, so the program should ideally show you on the screen how your letter looks. But with Bank Street Writer, the text on the screen doesn't necessarily look like the printed text. Instead, the program asks you several on-screen questions about how you want the printed text to look; the results of your answers appear on the printout, not on the screen. Specifications you supply include the line width, line spacing (1, 2, or 3), page number for the first page of the file (in case it's a later section of a larger document), if you want a header and how you want it to read, and whether you want the computer to pause at the end of each page to let you change sheets, or to go on to the next page of your continuous form printing. If the appearance of the final printed letter doesn't please you, it is a simple matter to change your answers to the on-screen questions and print out another copy.

THE HISTORY ESSAY

Writing a research paper calls for not only a word-processing program, but a program that helps you organize your thoughts and perhaps helps you organize your research, as well. For a report about the early Roman emperors for a high school history class, I'm going to use AppleWorks, an integrated productivity program.

Integrated software is going around as a popular computer buzzword today. It means that two or more

productivity programs are linked together into one larger program. The modules share similar commands, decreasing the difficulty in learning several different programs at once. Moreover, information from one module can be carried over to another. AppleWorks combines three modules—word processing, data-base management, and electronic spreadsheet—into a single program. We'll have much more to say about electronic spreadsheets in the next chapter. But for my history assignment, I'll be using the word-processing and data-base modules.

The first step in preparing for the paper is doing research. Since the topic is the early Roman emperors, I'll set up a data base to help organize information about each emperor. A data base is nothing more than a collection of information. In its broadest sense, an example of a rather large and diverse data base is a public library. But in computer jargon, a data base is more narrowly defined, like a listing of Roman emperors and pertinent facts about them. (We'll have much more to say about data bases in the chapter on electronic filing.)

Figure 5-1 shows a typical screen from my data base called Emperors. Earlier, I had used AppleWorks to define the categories (Emperor, Reign, Origin, etc.), and put them into an on-screen form I can fill out every time I have information to add about one of the emperors. The display shows that I have records for a total of nine emperors, and that I'm currently displaying record number one.

To get a bigger picture of the information stored in my Emperor data base, we can look at a screen listing of

```
File: Emperors                 REVIEW/ADD/CHANGE             Escape: Main Menu

Selection: All records


Record 1 of 9
================================================================================
Emperor: Augustus Caesar
Reign: 27 BC - 14 AD
Origin: -
Conquests: Egypt; Spain;to Danube River
Laws: Encouraged marriage; Right of 3 Children;Divorce penalties
The Arts: Vergil(The Aeneid);Horace(Odes);Ovid(Metam.);Livy(History)
Downfall-: Died of natural causes
Misc #1: Decreed a god in 1 AD
Misc #2: Also called Octavian
Misc #3: -




--------------------------------------------------------------------------------
Type entry or use @ commands                                          @-? for Help
```

Figure 5-1. **Sample record from a history paper research data base. On the video screen, the "@" symbol appears as an Open Apple sign.**

```
File: Emperors              REVIEW/ADD/CHANGE            Escape: Main Menu

Selection: All records


Emperor          Reign          Origin          Conquests       Laws
=========================================================================
Augustus Caesar 27 BC - 14 AD   -               Egypt; Spain;to Encouraged marr
Tiberius         14-37          -               Status Quo      -
Gaius (Caligula 37-41           -               unsuccesful cam -
Claudius         41-54          Lyons           Britain;2 provi -
Nero             54-68          -               -               -
Galba            68-69          -               -               instituted unpo
Otho             69             -               -               -
Vitellius        69             -               -               -
Vespasian        69-79          Sabine Reate    Northern Englan -



-------------------------------------------------------------------------
Type entry or use @ commands                                 @-? for Help
```

Figure 5-2. **Master listing of all records in a data base.**

all records on file (Figure 5-2). The screen isn't wide enough to show every category for each emperor, but at least I can see which emperors are listed. When I place the cursor on the line of a particular record I want to see, I press a two-key command to see the full record like the one in Figure 5-1.

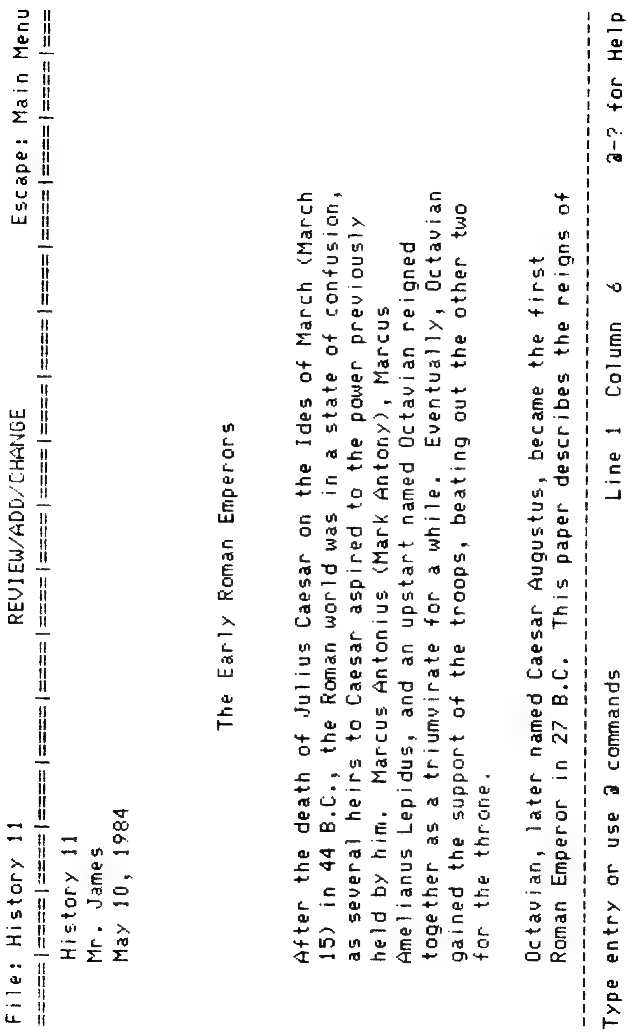
Once all the research is done, I can flip through the records one by one, to get an overview of the information collected. Then I can use the word-processing module to create an outline of the essay (Figure 5-3). One nice feature about AppleWorks is that I can have several documents active in the //c memory at one time. Recalling a document to the screen entails only a couple of keystrokes. Therefore, once I finish the outline, I can constantly refer back to it as I write the actual essay. Wherever I left the cursor in the outline, I'll return to that very spot the next time I want to look at it.

Figure 5-4 shows the word-processing screen with the beginning of the first draft of my essay. Notice how the title is properly centered. AppleWorks offers a large library of special page appearance options, which are selectable with only a couple of keystrokes.

Let's say I want to present as a part of the essay a table of the emperors covered in this report, along with a few pertinent facts pulled from the data base. To do that, I go back to the Emperor data base and give the command to initiate a report format—how I want the information I'm going to pull from the data base to look when I place it in the table. Figure 5-5 shows the screen used to select the categories "Emperor," "Reign," and "Downfall" and to determine how wide each column is to be. From this point, I issue the

```
File: Emperor Outline              REVIEW/ADD/CHANGE              Escape: Main Menu
=====|====|====|====|====|====|====|====|====|====|====|====|====|====|====|===
May 5, 1984

             The Early Roman Emperors -- Outline

I.    Introduction

II.   Augustus
      A. His battles with Mark Antony for power.
      B. Events leading to his being named Emperor
      C. Military campaigns for consolidation of frontiers
      D. Encouragement of the arts.
         1. Poets
            a. Virgil
            b. Tibullus
               c. Propertius
         2. Essayists
            a. Ovid
            b. Seneca

III. Tiberius
--------------------------------------------------------------------------------
Type entry or use @ commands          Line 1   Column 1           @-? for Help
```

Figure 5-3. **You can use a word processing program to create an outline.**



```
File: History 11          REVIEW/ADD/CHANGE          Escape: Main Menu
====|====|====|====|====|====|====|====|====|====|====|====|====|====
   History 11
   Mr. James
   May 10, 1984

                    The Early Roman Emperors

After the death of Julius Caesar on the Ides of March (March
15) in 44 B.C., the Roman world was in a state of confusion,
as several heirs to Caesar aspired to the power previously
held by him.  Marcus Antonius (Mark Antony), Marcus
Amelianus Lepidus, and an upstart named Octavian reigned
together as a triumvirate for a while.  Eventually, Octavian
gained the support of the troops, beating out the other two
for the throne.

Octavian, later named Caesar Augustus, became the first
Roman Emperor in 27 B.C.  This paper describes the reigns of
--------------------------------------------------------------------
Type entry or use @ commands       Line 1  Column 6      @-? for Help
```

Figure 5-4. **Sample screen of the AppleWorks word processing module.**

command that tells the *//c* to print the report as defined by the format. The program gives us a number of options as to where I want the report printed (Figure 5-6), either to the printer, to the screen, to a feature called the clipboard, or to a special disk file for use with other programs. I'm going to choose the clipboard.

Mimicking the clipboard feature of the Apple Macintosh computer, the AppleWorks clipboard is a kind of repository where you temporarily store things so you can transfer them over to one of the other program modules. When I select the clipboard option in the screen shown in Figure 5-6, the program makes a copy of the report according to the format design onto the clipboard.

To get the table into the essay, I return to the essay (only a couple of keystrokes, with plenty of on-screen help), and place the cursor where I want to move the information from the clipboard to the document. The Move command asks whether I want to move something within the same document, from the current document to the clipboard, or from the clipboard to the current document. The last choice is the one I want here. Literally at the press of one more button, the table is inserted into my essay, as shown on the screen in Figure 5-7.

Since the information about Claudius' downfall is longer than the space I allowed for that category, I'll have to move the cursor up to the parenthesis after Agrippina and delete a few characters. I do this by invoking the Delete command, at which point I'm asked to move the cursor to mark the text to delete. As I move the cursor, the text behind it becomes high-

```
File: Emperors                     REPORT FORMAT              Escape: Report Menu
Report: Reigns
Selection: All records



==================================================================================
--> or <--  Move cursor                   ∂-J  Right justify this category
 > ∂  <     Switch category positions     ∂-K  Define a calculated category
--> ∂ <--   Change column width           ∂-N  Change report name and/or title
∂-A  Arrange (sort) on this category      ∂-O  Printer options
∂-D  Delete this category                 ∂-P  Print the report
∂-G  Add/remove group totals              ∂-R  Change record selection rules
∂-I  Insert a prev. deleted category      ∂-T  Add/remove category totals
----------------------------------------------------------------------------------

Emperor                Reign          Downfall-                        L
-A-------------------- -B----------- -C---------------------------    e
Augustus Caesar.       27 BC - 14 AD Died of natural causes            n
Tiberius               14-37         Died of natural causes            6
Gaius (Caligula)       37-41         Assasinated                       7

----------------------------------------------------------------------------------
Use options shown above to change report format                       50K Avail.
```

Figure 5-5. **On-screen guidance helps you create a report format.**

```
File: Emperors                PRINT THE REPORT          Escape: Report Format
Report: Reigns
Selection: All records
================================================================================
  Where do you want to print the report?

  1.  Apple DMP

  2.  The screen

  3.  The clipboard (for the Word Processor)

  4.  A text (ASCII) file on disk

  5.  A DIF (TM) file on disk

--------------------------------------------------------------------------------
Type number, or use arrows, then press Return                          50K Avail.
```

Figure 5-6. **You can print a report to the printer, screen, program clipboard, or to the disk as files for other programs.**

```
File: History 11              REVIEW/ADD/CHANGE             Escape: Main Menu
=====|====|====|====|====|====|====|====|====|====|====|====|====|====|====|===
     for the throne.

     Octavian, later named Caesar Augustus, became the first
     Roman Emperor in 27 B.C.  This paper describes the reigns of
     the first emperors to rule the Roman Empire, focusing
     especially on how so many of them met their ends at the
     hands of others.  Here is a list of the emperors discussed:

Emperor                 Reign         Downfall-
----------------------- ------------- ----------------------------
Augustus Caesar         27 BC - 14 AD Died of natural causes
Tiberius                14-37         Died of natural causes
Gaius (Caligula)        37-41         Assasinated
Claudius                41-54         poisoned by Agrippina(not co
Nero                    54-68         suicide
Galba                   68-69         assasinated after 4 months
Otho                    69            suicide after 4 months
Vitellius               69            assasinated after 8 months
Vespasian               69-79         natural causes

-------------------------------------------------------------------------------
Type entry or use @ commands            Line 19  Column  1         @-? for Help
```

Figure 5-7. **AppleWorks lets you integrate information from a data base into a word processing document.**

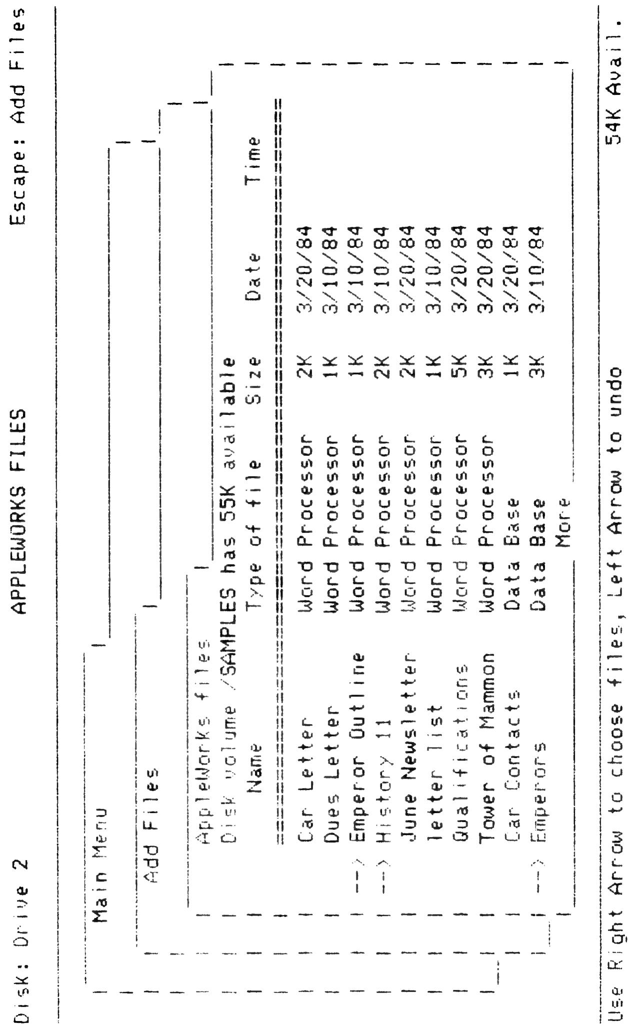


Figure 5-8. **Visual aids help the user find the way through program commands.**

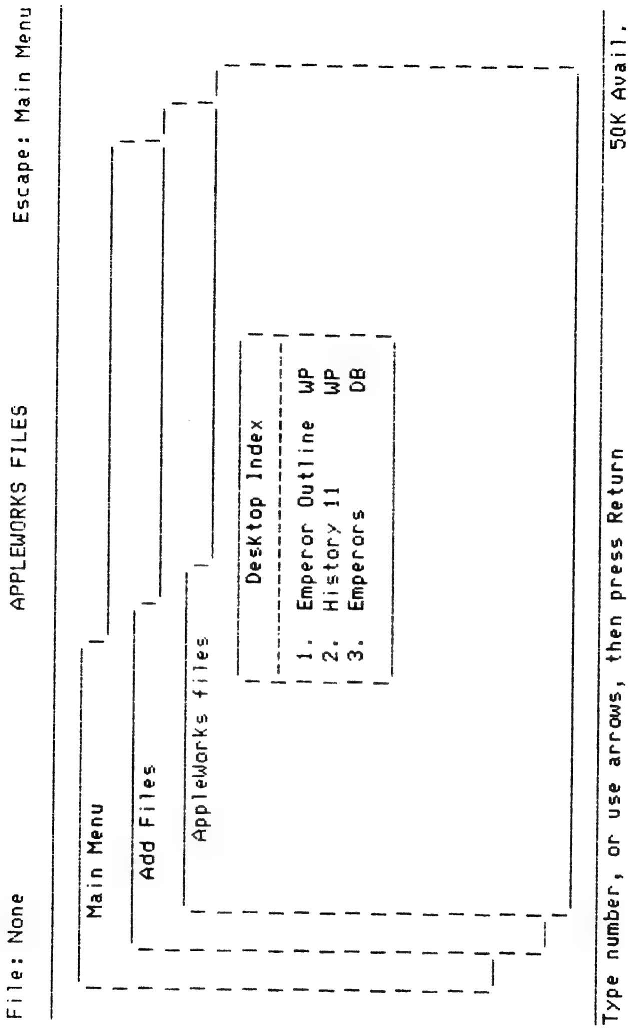


Figure 5-9. **Several AppleWorks documents can be in active memory at one time, each retrievable with a couple key-strokes.**

lighted in inverse video (black letters on white background). When I get to the end, I simply press the Return key, and the extraneous text is removed.

If I were interrupted while writing the essay, I would save everything up to that point on the disk. The next time I start the AppleWorks program to do some more work on this project, I can bring all three documents into active memory at once. The screen in Figure 5-8 shows how I've selected the three documents I need to work with for the essay: Emperor Outline (my essay outline); History II (the essay itself); and Emperors (the data base containing the results of my research). I select which document to work on by typing in the number of the document listed in the Desktop Index, shown in Figure 5-9.

Another program along these lines worth investigating is called Think Tank (Living Videotext). Although it doesn't have the data-base and report functions of AppleWorks, it does help you organize information particularly well in outline form. You can create outlines with several levels of subheadings quite easily. Then, if you want to see only the major headings to see how the overall train of thought flows, you can collapse all the subheadings on the screen (although they're still in memory). Its designers call the program an idea processor rather than a word processor, since it helps you organize your thoughts for just about any situation.

THE PERSONALIZED FORM LETTER

You've probably received advertisements in the mail that have been customized for the name and address printed on the mailing label. If your name is Sally

Smith and you live in Kansas City, one of the letters inside the mailing will have a statement like, "You may have already won $1,000,000, which we'll deposit for you in the Smith family savings account at your favorite bank near your Kansas City home." I don't know if anyone ever really believes that someone actually types this kind of letter; the truth is, of course, that this kind of mailing uses a computer that combines two programs: word processing and data base. If you're an officer of a church group or other organization, you can do a similar kind of mailing with an Apple //c, while keeping your typing to a minimum.

Here's a typical situation. Your club has just decreed that all members who joined the club before 1976 will receive a reduced rate for the next year's annual dues. Of course you could send a photocopied notice to everyone, but it would be a nice touch to send a personalized note to each of the members who qualify. You're going to use the data-base module from AppleWorks to find those members who joined before 1976, and then use the AppleWorks word-processing module to send a letter to each one of them. To do this, you will create the master letter and then insert into each letter the names, addresses, and the date each member joined the club.

This presumes, of course, that you've been keeping the membership roster on the //c with AppleWorks all along. Figure 5-10 shows a typical record for an individual member, with each of the categories of information filled in. Figure 5-11 shows all the records in your file called Member Roster. By moving the cursor to any one of the records in Figure 5-11 and typing the Zoom command (Open Apple-Z), you see the record in full.

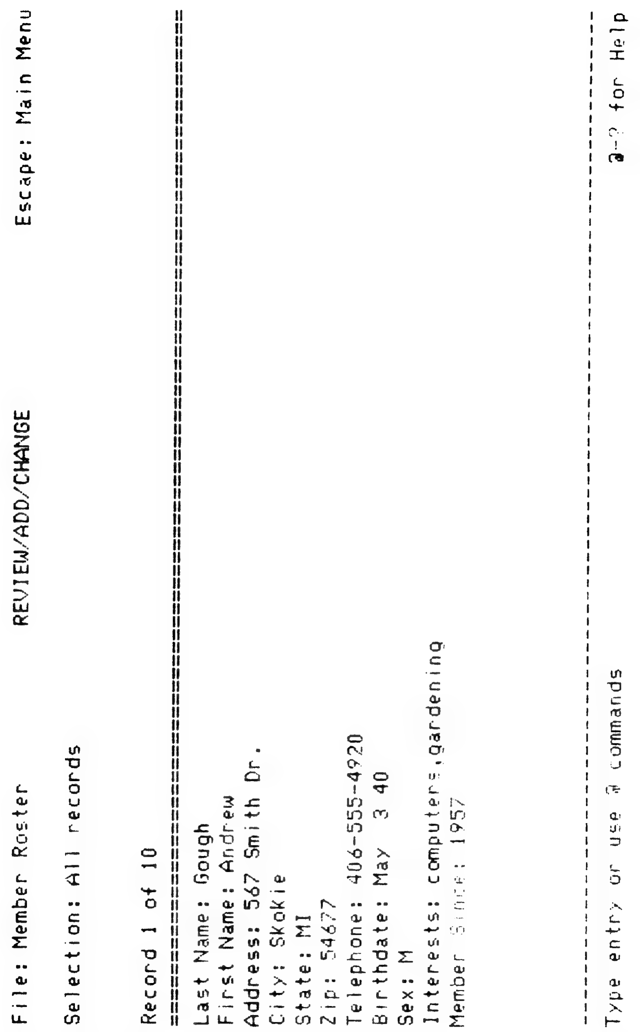


Figure 5-10. **Sample record from a club member data base record.**

```
File: Member Roster             REVIEW/ADD/CHANGE            Escape: Main Menu

Selection: All records

Last Name        First Name       Address           City              State
===============================================================================
Gough            Andrew           567 Smith Dr.     Skokie            MI
North            Leslie           12059 S. Lombar   San Gregorio      WY
Hutchinson       Timothy          RRte 3            Mt. Pleasant      PA
DeVane           Simon            54 Burton Ave.    Portland          GA
Sagara           Denise           899 DaNova Lane   Lancaster         TX
Whitmore         Samuel           1239 E. 34th St   New York          NY
Reed             Donald           665 Ocean Blvd    Santa Monica      CA
Bartelson        Lydia            7859 Greenwood    Omaha             NE
Haas             Helmut           555 Lake Shore    Toronto           ONT
Rogers           Steven           900 Chestnut Dr   Chicago           IL



-------------------------------------------------------------------------------
Type entry or use @ commands                                      @-? for Help
```

Figure 5-11. **Full on-screen data base list of all club members.**

Utilizing a few commands within the data base, you can select from all the records only those who were members before 1976. Note in Figure 5-10 that one of the categories of information is the year each member joined. All you have to do is have the program search through every record and pull out those whose "Member Since" category is less than 1976 (menus of these options appear on the screen to guide you along).

Next you use the Report function of the data-base module to produce a report with the names and addresses already in the proper format for inserting at the head of a letter. You also want to bring over the year each member started. As I did with the emperor report in the history essay, you print the report to the clipboard (Figure 5-6). This time, however, you're going to do something different with the information now stored in the clipboard. To make the qualifying members' data handy for movement over to your canned letter, you will transfer the contents of the clipboard to a temporary word-processing document. To do that, start up a new word-processing document, called "letter list," and copy the information from the clipboard to the document (Figure 5-12).

The next step is to create your canned letter, most of which shows up on the screen illustrated in Figure 5-13. Notice that in the next-to-last line of the first paragraph, there is an extra space to the left of the comma. This is where you will insert the year the member joined. Going back to "letter list," you issue the command to move the first member's information over to the clipboard. Switching over to the dues letter, you then move the data from the clipboard into the top of the letter (Figure 5-14). Notice that the date

```
File: letter list                REVIEW/ADD/CHANGE               Escape: Main Menu
=====|====|====|====|====|====|====|====|====|====|====|====|====|====|====|===
File:   Member Roster
            Page  1
Report: list
Selection: Member Since is less than 1976

Andrew Gough
567 Smith Dr.                  1957
Skokie MI 54677



Timothy Hutchinson
RRte 3                         1975
Mt. Pleasant PA 32245



Helmut Haas
555 Lake Shore Drive           1975
Toronto ONT 5B3 403
-------------------------------------------------------------------------------
Type entry or use @ commands          Line 1  Column  1          @-? for Help
```

Figure 5-12. **Report of selected members for use in merging with a personalized letter.**

```
File: Dues Letter              REVIEW/ADD/CHANGE              Escape: Main Menu
=====|====|====|====|====|====|====|====|====|====|====|====|====|====|====|===
June 1, 1984



Dear :

According to a change in the organization's bylaws, all
members in good standing who have belonged to the club for 8
years or more are entitled to a reduction in annual dues.
Since you have been a member since , you are entitled to
this reduction.

Effective with your next dues notice, the annual charges
will go down from $15 to $12 per year.  I know you'll be
please with this change for our long-time members.

If you have any questions about this, please feel free to
write me.  We look forward to receiving more contributions
from you for our quarterly newsletter.

-------------------------------------------------------------------------------
Type entry or use @ commands            Line 1  Column  1            @-? for Help
```

Figure 5-13. **The blank letter into which names, addresses, and other information will be moved.**

```
File: Dues Letter                REVIEW/ADD/CHANGE                Escape: Main Menu
=====|====|====|====|====|====|====|====|====|====|====|====|====|====|====|===
June 1, 1984


Andrew Gough
567 Smith Dr.                    1957
Skokie MI 54677




Dear :

According to a change in the organization's bylaws, all
members in good standing who have belonged to the club for 8
years or more are entitled to a reduction in annual dues.
Since you have been a member since , you are entitled to
this reduction.

Effective with your next dues notice, the annual charges
will go down from $15 to $12 per year.  I know you'll be
-------------------------------------------------------------------------------
Type entry or use @ commands          Line 4  Column  1          @-? for Help
```

Figure 5-14. **Bringing one of the records from Figure 5-12 into the blank letter document.**

June 1, 1984

Andrew Gough
567 Smith Dr.
SKokie MI 54677

Dear Andrew:

According to a change in the organization's bylaws, all members in good standing who have belonged to the club for 8 years or more are entitled to a reduction in annual dues. Since you have been a member since 1957, you are entitled to this reduction.

Effective with your next dues notice, the annual charges will go down from $15 to $12 per year. I know you'll be please with this change for our long-time members.

If you have any questions about this, please feel free to write me. We look forward to receiving more contributions from you for our quarterly newsletter.

Yours truly,

Sarah Procter
Secretary

***Figure 5-15.* The finished printed letter, with address, salutation, and starting date inserted.**

is a separate entry off to the right. All you have to do is move it down to the spot reserved for it, type in the name at the salutation, and print the letter.

For each succeeding member, just delete the current name and address, and move the next member from "letter list" by way of the clipboard. By moving the data out of "letter list," instead of copying it, you are assured that every member in "letter list" will get a letter. Otherwise, you might lose track of which names already had a letter done for them. This way, when the "letter list" is empty, you know every qualifying

member received a letter. The final letters look like the one shown in Figure 5-15, and can be printed on club stationery with the Imagewriter or Scribe printer attached to the *II*c.

These examples of writing with an Apple *II*c may seem specialized to you. They were presented, however, to give you a feeling for how the *II*c–turned–word processor can be a productive tool for all writing tasks. Other word-processing programs let you perform form letter merging more automatically, while others make it easy to print footnotes automatically at the bottom of pages for academic papers. Your writing may also entail the sending of documents to another computer across the country. If so, take a look at how you can do this in Chapter 8. You can also illustrate your work with a program like MousePaint, which is described in detail in Chapter 9. Writing encompasses so many different disciplines that we cannot cover them all here—I'm sure, however, that you have thought of several of your own special ways of using a word processor already.

And remember, when shopping for word-processing and related software, avoid programs that require you to devote a great deal of attention to the operating of the program and commands. Instead, seek out those programs that free your mind to concentrate on your writing, not your computing.

CHAPTER 6

The Apple *//c* as a Number Cruncher

Computers are at their best when working with numbers. The earliest computers were created for no other reason than to handle numbers. Complex military calculations were among the first applications for giant room-sized computers of the 1940s. Even today, personal computers operate at their fastest when manipulating—crunching—numbers for their users.

In this chapter, we'll be looking at two of the most popular ways of working with numbers on the *//*c. One involves a most flexible kind of program, called an electronic spreadsheet, which organizes all kinds of figures, from bean counts to dollars. The other *//*c number environment we'll examine is a more specialized one for the management of personal budgets and investment portfolios—programs that deal exclusively with money.

ELECTRONIC SPREADSHEETS

In the early days of personal computers, the question of what someone could do with a computer was a question with fewer realistic answers than today. Unless you knew how to program your shiny new computer to perform the kind of magic you expected, there was little more you could do with it than play one of the hundreds of games on the market. Then came a program that changed it all, and helped make the Apple *//* computer a much more popular accoutrement of daring business executives. That program was VisiCalc, the first commercial electronic spreadsheet. Today, VisiCalc and dozens of its descendants are among the first programs added to a personal computer owner's software library.

An electronic spreadsheet is more than a mere video representation of a columnar pad for entering numbers. It is a powerful calculation tool as well. This is because if you change one or more of the figures in any of the columns or rows, the computer constantly recalculates the subtotals, totals, averages, percentages, and other functions scattered about the sheet. When you make a change of one value, all other values affected by it anywhere in the spreadsheet are instantaneously adjusted. No longer do you have to punch all the numbers in the calculator every time you make some changes to one of the figures, nor do you ever again have to double-check your addition.

The size of an empty spreadsheet on the *//*c depends on the spreadsheet program you're running. One of AppleWorks' modules is a spreadsheet program capable of holding roughly 6000 boxes of figures—called

cells—in a single spreadsheet. The reason for such an enormous spreadsheet is that many experienced spreadsheet users combine a few related spreadsheet assignments on one large spreadsheet, so they don't have to carry over or repeat entries from sheet to sheet. For example, if the boss wants quarterly budget reports, plus an annual summary, you can keep the four quarterly figures and the annual summary separated by using different areas of the overall spreadsheet.

Creating a spreadsheet is rather simple once you've done it once or twice. A blank spreadsheet, like the AppleWorks one illustrated in Figure 6-1, may seem rather intimidating, but think of it as a video version of a columnar pad. To it you add labels down the left margin (like budget categories) and across the top (like months of the year). And, as on the paper version, you can have subtotal and total columns and rows placed throughout to help you categorize groups of figures (e.g., Net Profit/Loss, Six Months Total, etc.). Each intersection of a column (designated by the letters across the top) and row (numbered along the left margin) is called a cell. Each cell has an address made up of its column and row locations. Therefore, the cell in the upper left corner of Figure 6-1 is called cell A1. Two cells down the column, the cell is called A3. A cursor on the screen highlights the cell wherever it's located, and the cell location of the cursor is shown at the lower left corner of the screen (A1 in Figure 6-1). Cursor movement is controlled primarily by the cursor keys on the *II*c keyboard.

To enter labels, you move the cursor to the cell in which you want to begin, and start typing letters.

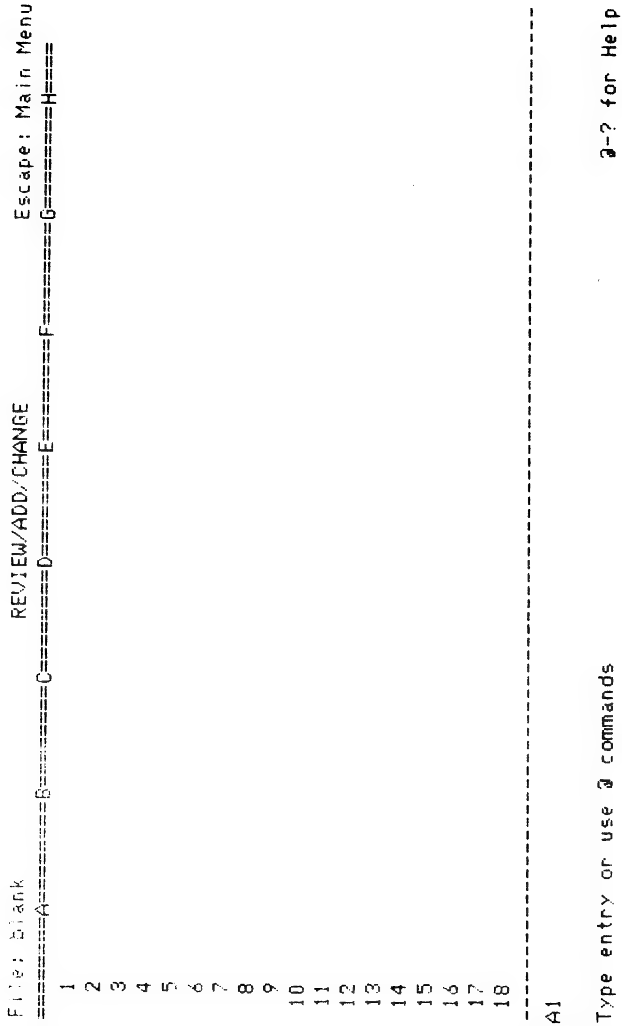


Figure 6-1. **Empty spreadsheet from AppleWorks.**

Move the cursor to other cells anywhere on the spreadsheet to put in more labels.

In Figure 6-2, the screen shows the results of a personal net worth analysis (from the AppleWorks sample data disk). Since, like a word-processing document, a spreadsheet may be much larger than what you see on the screen, don't assume that you are looking at the whole thing. The asset labels, for example, really start in column B, as you can see by the partial labels showing in rows 49 and 52.

Next you want to put formulas in those cells that perform calculations on other cells. Typically, the formulas can be quickly typed in with the help of built-in functions that perform addition (for totals), subtraction (for net figures), averages, and percentages. In Figure 6-2, for example, the asset figures are totaled in column E, while liabilities are totaled in column J. AppleWorks also has built-in functions that calculate net present value and make decisions based on the values in certain cells (if . . . then decision making). You can see an example of a formula and a built-in math function (sum) in Figure 6-2, where the cursor marker in the lower left corner of the screen shows what is in cell F52. The function is summing the difference between cells E49 (current assets) and J49 (current liabilities) to arrive at the net worth displayed in cell F52.

With AppleWorks, you can not only print out your spreadsheets by themselves (leaving off the row and column markers that appear on the screen), but you can also copy figures to the clipboard for inclusion in word-processing documents. This is particularly

```
File: Personal Worth              REVIEW/ADD/CHANGE              Escape: Main Menu
========C========D========E========F========G========H========I========J====
 36 Clothing                4500
 37 Jewelry, Tableware      1250
 38 Computer Equipment      2750
 39 Stamp Collection        1250
 40                    ---------
 41                        32750
 42
 43
 44 Tax refund due           375
 45 Loan to Bill             900
 46                    ---------
 47                         1275
 48                    ---------                                    ---------
 49 RENT ASSET VALUE      468760      TOTAL CURRENT LIAB. VALUE       180375
 50                    =========                                    =========
 51
 52  AS OF THIS DATE                 288385  <<<<<<
 53 ====================================
--------------------------------------------------------------------------------
F52: (Value, Layout-F0) @SUM(E49-J49)

Type entry or use @ commands                                      @-? for Help
```

Figure 6-2. **A spreadsheet can extend beyond the screen's window.**

handy if you are preparing a report or correspondence that relies on figures derived from the spreadsheet to make a point. Instead of referring the reader to an attachment, you can include pertinent spreadsheet columns and rows right in the middle of the document where it will have the greatest impact.

To simplify the creation process of a useful spreadsheet, some spreadsheet software, notably Multiplan (Microsoft), has predesigned spreadsheet forms that you can load immediately into the *II*c and start filling in with figures. These prewritten spreadsheet forms are called *templates*, and Microsoft includes two in the *II*c version of Multiplan for loan calculations and household budgeting. Likewise, books about Multiplan and other spreadsheet programs are available that include many predesigned forms. All you do is type in the values and formulas according to carefully prepared lists of what goes in each cell. Since there is a lot of similarity in math function commands in the popular spreadsheet programs, you can readily adapt a template designed for a program other than the one you use.

A spreadsheet is probably one of the most versatile programs you'll ever run on the *II*c. It can be adapted to so many different applications that it's unlikely anyone has cataloged them all. We'll see in the chapter describing the *II*c as an educational tool how a teacher can use a spreadsheet to track and calculate grades from pupils' assignments. Any kind of small business or organization that does bookkeeping can readily adapt its entire system to a spreadsheet to track all the accounting. Many executives and field salespeople who don't have access to the corporate computer

often use a computer like the *//c* equipped with a spreadsheet program to track performance and produce budget forecasts on their own time, often at home.

THE SALES LOG

Let's create a spreadsheet that a salesperson might use to track unit and dollar sales of each product in the company's line. To simplify our displays here, we'll limit the extent of our tracking to three months. First we'll make the units sales chart, and then the dollars chart.

In Figure 6-3, we've already written out all the labels, including a heading across the top, so we know precisely what figures we're looking at. At the right are two kinds of totals columns. One of them, Total Qtr., adds up the total number of units sold for each product. The last column, % of total, calculates the percentage of the total number of units that each product represents. With this figure, we'll be able to track our product mix. We could have easily created a similar calculation for each month to see how the product mix might change on a monthly basis. In this illustration, figures have been entered only for July and August.

Figure 6-4 is exactly the same spreadsheet, except that it calculates in dollars, not units, and it starts below the first spreadsheet, on line 18. In each cell where a figure is located, a formula was entered when this segment of the spreadsheet was created. The formula for cell C22, for example, reads +C8*25. In other words, the number that appears in cell C22 is the value of what is in cell C8 (the number of units sold) multi-

```
File: sales.product            REVIEW/ADD/CHANGE               Escape: Main Menu
========A========B========C========D========E========F========G========H====
  1
  2                     Quarterly Sales by Product (Units)
  3 ========================================================================
  4                     July      August    September         Total Qtr % of tot
  5
  6 Product Name:
  7
  8 Widget                    12        15          0                27    25.5%
  9 Wodget                    20        21          0                41    38.7%
 10 Blodget                    5         4          0                 9     8.5%
 11 Blunget                    4         9          0                13    12.3%
 12 Blunget Deluxe             1         2          0                 3     2.8%
 13 Super Blunget              3         2          0                 5     4.7%
 14 Blunget Acc. Kit           3         5          0                 8     7.5%
 15
 16
 17
 18                     Quarterly Sales by Product (Dollars)
------------------------------------------------------------------------------
E1

Type entry or use @ commands                                         @-? for Help
```

Figure 6-3. **Spreadsheet created to track sales of products by Units.**

```
File: sales.product            REVIEW/ADD/CHANGE            Escape: Main Menu
========A========B========C========D========E========F========G========H====
 17
 18                 Quarterly Sales by Product (Dollars)
 19 =======================================================================
 20                 July     August   September            Total Qtr % of tot
 21
 22 Widget               300      375        0                  675    13.6%
 23 Wodget               620      651        0                 1271    25.7%
 24 Blodget              250      200        0                  450     9.1%
 25 Blunget              180      405        0                  585    11.8%
 26 Blunget Deluxe        95      190        0                  285     5.8%
 27 Super Blunget        315      210        0                  525    10.6%
 28 Blunget Acc. Kit     435      725        0                 1160    23.4%
 29
 30
 31 Total               2195     2756        0                 4951   100.0%
 32
 33
 34
---------------------------------------------------------------------------
E34

Type entry or use @ commands                                   @-? for Help
```

Figure 6-4. **Continuation of spreadsheet in Figure 6-3 with sales by Dollars.**

```
File: sales.product             REVIEW/ADD/CHANGE             Escape: Main Menu
========A========B========C========D========E========F========G========H====
  8 Widget                    12       15        0                 27    17.9%
  9 Wodget                    20       21       45                 86    57.0%
 10 Blodget                    5        4        0                  9     6.0%
 11 Blunget                    4        9        0                 13     8.6%
 12 Blunget Deluxe             1        2        0                  3     2.0%
 13 Super Blunget              3        2        0                  5     3.3%
 14 Blunget Acc. Kit           3        5        0                  8     5.3%
 15
 16
 17
 18                Quarterly Sales by Product (Dollars)
 19 ================================================================
 20                July     August   September         Total Qtr % of tot
 21
 22 Widget                   300      375        0                675    10.6%
 23 Wodget                   620      651     1395               2666    42.0%
 24 Blodget                  250      200        0                450     7.1%
 25 Blunget                  180      405        0                585     9.2%
--------------------------------------------------------------------------------
E9: (Value, Layout-F0) 45

Type entry or use @ commands                                      @-? for Help
```

Figure 6-5. **Adding one figure to cell E9 causes instantaneous changes to several other cells throughout the spreadsheet.**

plied (an asterisk is the computer's way of signifying multiplication) by 25, which stands for the unit cost of $25.00. Therefore, every time we enter a figure in the Units spreadsheet above, the corresponding Dollars cell is automatically entered for us. Let's see how this is done.

We're going to enter our first figure for some Wodgets sold in September. Figure 6-5 shows the pertinent portions of both the Units and Dollars spreadsheets. When we type in the value, 45, for 45 Wodgets sold in September, several things happen automatically. First of all, the 45 units are added to the total Wodget units in cell G9, which in turn affects the percentage of total units sold in cell H9. Next, the dollar value of the 45 Wodgets is automatically entered in cell E23, and that figure is added to the total quarter Wodget dollars in cell G23, and the percentage of total dollar sales in cell H23. If we could see down to the bottom of the Dollars spreadsheet, we would see that the totals there have been adjusted accordingly. That's a lot of action for changing one little figure, but that's the kind of power available in a spreadsheet.

HOUSEHOLD BUDGETING

Sometimes, however, the flexibility of a spreadsheet creates too much work for someone who wants instant solutions without time-consuming setup. In the area of personal budget management, many software suppliers have come to the rescue. Some of these programs act like spreadsheets, while others add a great deal more to the program to make it much easier to use and much more specific in its ability to find an answer to a specific financial question you have, like how much

money you have to sock away each month if you're going to afford a college education for your two children in six and ten years, respectively.

The Financial Cookbook (Electronic Arts) is a brand-new program that approaches personal financial decisions from a realistic perspective, taking into account factors like your marginal tax rate and the inflation rate—factors most other programs ignore. You select from a menu of situations, such as living off an IRA, or a buy vs. lease decision for an automobile. From there, the program asks you to enter various figures. Then the program calculates the pertinent facts according to complex financial formulas. You can keep entering different figures to experiment with things like changing interest rates, maturity dates, and so on, until you find the right combination that answers your needs.

PORTFOLIO MANAGEMENT

One kind of number that most people pay close attention to is the value of their investments. Calculating the value of diverse stock-market holdings is certainly not the easiest thing to do with pencil, paper, and calculator. You can easily spend hours poring over the *Wall Street Journal* or *Barron's* to extract the figures you need. With the prospect of such a tedious job, it's not something you are prone to do on a frequent basis. But the *II*c can track these figures for you as often as you want, and with the kind of consistent accuracy that computers are known for.

The best way for the computer to gather stock-market quotations is for it to retrieve the information from a

national data base like the one Dow Jones News/Retrieval operates. To tap into this data base, you need a computer with the ability to communicate over telephone lines. The *//c* does just that with the addition of a telephone modem and software that turns the computer into a telecommunications terminal (see Chapter 3).

To facilitate the retrieval of stock information, Dow Jones offers a special software package for the *//c*, called the Dow Jones Investor's Workshop. Together with the computer and modem, the software dials the data base (via a local phone call in most cities), makes the inquiries about the holdings you have, disconnects, and then calculates the current value and gains or losses for the day. And because the Apple modem is an automatic-dial modem, the entire process takes only one command. You can go about other household chores while the computer makes the telephone call and data transfer. The computer won't know if you change your holdings, of course, so the one thing you do have to do is update the number of shares of each stock you have if you buy or sell.

One great advantage of having the computer make its inquiries from a predetermined list of stocks is that the computer can retrieve the information in much less time than if you were to call up the value of each stock yourself. Since you are charged by Dow Jones News/Retrieval for the length of time you are connected to the service, the computer software will save on connect charges. Once the information is stored in the computer, the connection is cut, and you can have the computer perform numerous calculations for you based on the fresh data.

TAX PLANNING

After you've made a killing in the stock market and figured your personal budget, it's time to use the *II*c for guidance in ways to reduce your income taxes. The Personal Tax Planner (Aardvark/McGraw-Hill) is a specialized number cruncher that lets you play "what if" questions about your personal finances as they affect your tax liabilities. In fact, you can list up to five different scenarios in adjacent columns to see how the bottom lines of each come out. The software has built into it the latest tax tables, so all you have to deal with are the income and expense figures for your household. Of course, no computer software can take the place of a good accountant, especially when the tax laws change, but software like this can give you a head start in figuring options based on personal financial decisions.

These are only a few ways the *II*c can help you manage numbers. You can be assured, however, that if your work, hobby, or community activities involve figures, there's the right *II*c software out there to help you, from keeping bowling team statistics to solving calculus problems.

CHAPTER 7

The Apple //c as Master Filer

Experienced computer users and computer salespeople are the first ones to throw around the jargon. When you talk to such people, you will soon hear the buzzwords for electronic filing: data-base management. But as we saw in our chapter about writing, a data base is nothing more complicated than a collection of information that can range in size from the number of cards on a Rolodex to the contents of the Library of Congress.

All of us collect information of some kind. Even the ragtag phone book of friends' numbers and addresses would be considered a personal data base. Infrequently used and small data bases don't necessarily need the power of a computer. In fact, as with balancing a checkbook, using a computer to hold your friends' phone numbers would probably be more time-consuming than keeping a little black book.

But occasionally, we have chores related to business, a hobby, or an outside organization that require not only the storage of information but frequent sorting,

arranging, and cross-referencing. For example, if you need to flip through dozens of file folders looking for the one that contains the scrap of information you need, then the *II*c might be able to hold the search time down from a half hour to a few seconds. Or if you have a collection of information that needs to be sorted in different ways from time to time or is frequently updated or extensively cross-referenced, the *II*c as a data-base manager stands to make your job much easier.

Basically speaking, a data-base program consists of four different subprograms: forms creation, information entry, record retrieval, and reporting.

The first step, forms creation, lets you design a form that you will ultimately use to enter data into the computer. If you've ever checked into a hospital or hotel that has a computer terminal set up at the registration desk, you will already be familiar with such an on-screen form. It's just like a paper form you fill out, except you type in the information after each request. Creating the form helps you organize your information collection before you start typing in the data.

Data entry is the most time-consuming work in building a data base. In fact, this is the step that is most often glossed over by computer marketers who tell you how easy life will be when you have instant access to information stored on your computer. If you already have a paper-based filing system, the most tedious job with a new computer—regardless of other applications you have for the machine—will be converting your paper files over to electronic form. Until such time as we can simply place documents in an optical character

reader that instantly converts printed text into computer data, we're going to be stuck with the keyboarding of data into the computer.

But what makes all that work worthwhile are the other powers of a data-base program. Once the information is stored on disk, you can load it into the computer and retrieve the records instantly. And a data-base program really pays its way when you have to track down records meeting several search criteria. For example, if you have a filing cabinet with customer files arranged alphabetically by last name, it might take days of looking through each file to pull out every customer who hasn't purchased an item over $100 within the last year. But with the customers set up in a *II*c data base, you simply specify that the entire data base is to be searched for customers meeting those criteria. In a matter of seconds, a list of those customers appears on the screen.

Computer control over a data base also lets you rearrange the files as you need them. One day, you may want to look at an alphabetical listing of customers. But if you would like to see your customers ranked by the amounts they've purchased, then the data-base program will sort them accordingly. Then, if you want to list them according to zip code for a mass mailing, you can do that at the press of a few keys.

The fourth subprogram of a data-base program is reporting. Most of the time, reporting takes the form of printed output for distribution to others, or for portability when you need to take information with you. Reporting is much more than just a printout of your regular data base, however. It lets you select

specific entries from your data base and arrange them in columns that make sense to the reader. Extraneous information can be left out. Moreover, some reporting programs include modest math capabilities so you can add up columns of figures, such as the total amount of sales or the total number of entries.

You can use the same data base to generate as many different kinds of reports as you want. The marketing department might want to see a report of all customers with their credit ratings and history of purchases. The sales promotion department might want the report to generate mailing labels of all your customers sorted according to zip code. Both report formats—a detailed customer report and mailing labels—pull information from the same base of information stored on disk.

MANAGING CLUB FILES

We've already seen how you can use some functions of the data-base module of AppleWorks in our chapter about writing. There, we merged information derived from a membership roster with a form letter to make the letters appear to be personalized. Refresh your memory about the information stored for each member in Figure 5-10 and how the program displays the master list of records in Figure 5-11.

The member records shown in Figure 5-11 are listed in the order in which they were entered into the computer—no particular order. But suppose we want to have the records in alphabetical order so that every time we call up the member roster to the screen, we can find a member's record faster. Also note that left

on its own, the program displays only five categories of information on the screen: last name, first name, address, city, and state. Let's rearrange the screen to show more information about each member without having to "zoom" up the entire record (even though zooming takes only a second, let's be lazy about it and let the computer do more of the work).

Rearranging the layout of the columns calls for the Layout command. A screen very similar to the one in Figure 5-5 appears, with the standard columns, headings, and a few sample records shown at the bottom of the screen. If you had an imaginary extra-wide video monitor extending far to the right, you could see columns for every item heading in your form. To bring as many columns as possible into view on our 80-column monitor, we use the arrow keys to expand or contract each column until the full width of the screen shows every column of information that we want (provided everything can fit within the width of one screen). Next, we want to alphabetize the listing on the main record screen.

To do this, we invoke the Arrange command, which presents the screen shown in Figure 7-1. If we had the cursor on any entry in the Last Name column when we called the arrange command, the instructions would tell us that the file will arranged by the Last Name category. At this point, we can tell the program which of four methods it should use to sort the records. Since our wish is for a traditional alphabetical listing, we'll select From A to Z. Instantly, the screen changes to the listing of records shown in Figure 7-2. Our records are now arranged in alphabetical order of the last names.

```
File: Member Roster              ARRANGE (SORT)              Escape: Erase entry

Selection: All records


================================================================================

                  This file will be arranged on
                  this category:  Last Name

                  Arrangement order:

                        1.  From A to Z
                        2.  From Z to A
                        3.  From 0 to 9
                        4.  From 9 to 0



--------------------------------------------------------------------------------
Type number, or use arrows, then press Return  3                      4K Avail.
```

Figure 7-1. **Choices in arranging a data base listing.**

File: Member Roster REVIEW/ADD/CHANGE Escape: Main Menu

Selection: All records

Last Name	First Name	Address	City	Stat	Zip	Telephone
Bartelson	Lydia	7859 Greenwood Te	Omaha	NE	70944	703-555-2462
DeVane	Simon	54 Burton Ave.	Portland	GA	33455	414-555-9483
Gough	Andrew	567 Smith Dr.	Skokie	MI	54677	406-555-4920
Haas	Helmut	555 Lake Shore Dr	Toronto	ONT	5B3 403	709-555-6933
Hutchinson	Timothy	RRte 3	Mt. Pleasant	PA	32245	412-555-0845
North	Leslie	12059 S. Lombard	San Gregorio	WY	87009	702-555-9766
Reed	Donald	665 Ocean Blvd	Santa Monica	CA	90445	213-555-7990
Rogers	Steven	900 Chestnut Driv	Chicago	IL	60644	312-555-6931
Sagara	Denise	899 DaNova Lane	Lancaster	TX	75660	214-555-7890
Whitmore	Samuel	1239 E. 34th St	New York	NY	10045	212-555-1212

Type entry or use @ commands @-? for Help

Figure 7-2. **Membership list sorted alphabetically by name.**

```
File: Member Roster            REVIEW/ADD/CHANGE             Escape: Main Menu

Selection: All records

Last Name   First Name Address            City          Stat Zip     Telephone
==============================================================================
Haas        Helmut     555 Lake Shore Dr  Toronto       ONT  5B3 403 709-555-6933
Whitmore    Samuel     1239 E. 34th St    New York      NY   10045   212-555-1212
Hutchinson  Timothy    RRte 3             Mt. Pleasant  PA   32245   412-555-0845
DeVane      Simon      54 Burton Ave.     Portland      GA   33455   414-555-9483
Gough       Andrew     567 Smith Dr.      Skokie        MI   54677   406-555-4920
Rogers      Steven     900 Chestnut Driv  Chicago       IL   60644   312-555-6931
Bartelson   Lydia      7859 Greenwood Te  Omaha         NE   70944   703-555-2462
Sagara      Denise     899 DaNova Lane    Lancaster     TX   75660   214-555-7890
North       Leslie     12059 S. Lombard   San Gregorio  WY   87009   702-555-9766
Reed        Donald     665 Ocean Blvd     Santa Monica  CA   90445   213-555-7990



------------------------------------------------------------------------------
Type entry or use @ commands                                      @-? for Help
```

Figure 7-3. **Membership list sorted according to zip code.**

Our next task is to prepare mailing labels for the mailing of the quarterly newsletter. The post office can distribute the newsletters faster when they are arranged in order of zip code. Before we print the labels, we rearrange the records by moving the cursor to any one of the zip code listings in Figure 7-2 and invoke the Arrange command. We respond to a screen similar to the one in Figure 7-1 by asking for zero through nine sorting. In a second, the computer produces the records in zip code order, as shown in Figure 7-3.

When it's time to print the labels, the program asks us if we want to create a new "report format," which will select which information we want to print, and how the printed output should be arranged. With AppleWorks, you create a report format by starting out with a listing of all categories for this file. The listing looks like the record shown in Figure 5-10 but without the entries—just the bare category names. Using the cursor movement keys and a couple of other commands on the *II*c keyboard, you delete the categories you don't want to print and move the ones you do want into the proper sequence. In the case of our mailing labels, we want the first line of the label to read like the format shown in Figure 7-4. The left arrows next to some categories tell the computer that, no matter how long the entries are, the ones to the right should be spaced out accordingly.

Note that the dotted line below the report form indicates the total number of lines that will be printed with the current format. In this case, three extra blank lines are included, bringing the total to six lines per record. The reason for this is that the single-column mailing labels we use on the printer come in continuous sheets

```
File: Member Roster                REPORT FORMAT               Escape: Report Menu
Report: mailing
Selection: All records

================================================================================
First Name <Last Name
Address
City <State <Zip

------------------------Each record will print  6 lines-------------------------



--------------------------------------------------------------------------------
Use options shown on Help Screen                                      @-? for Help
```

Figure 7-4. **Report format for printing mailing labels.**

with a center-to-center measurement equal to six printed lines. Therefore, each record will print on three of those lines, the printer will advance three more blank lines to the next label, and begin printing again precisely where we want it to.

DATA ANALYSIS

AppleWorks has a powerful search capability that goes beyond our examples. Its retrieval ability can be a valuable tool for analysis of information stored in the computer. For instance, an insurance agent might want to search through all client files and find only those clients who meet several different conditions, such as falling into a specific age range, with two or more children, and with a minimum income level. When you command the *II*c to make a search, you can specify several conditions that must be met (some programs make you search for each one individually, gradually narrowing down your list after three searches). Figure 7-5 shows you the options available to you every time you wish to make a search with AppleWorks. You can link as many as three conditions together to speed your search.

RESEARCH AND BIBLIOGRAPHIES

When I put together a data base for the history paper research in Chapter 5, you may have noticed that we were limited in the amount of information we could store with each item heading (for example, Emperor, Reign, etc.). If you need to store more information with each item heading, you may want to use pfs:File and its sister reporting program, pfs:Report (Software Publishing).

```
File: Member Roster          SELECT RECORDS          Escape: Review/Add/Change

Selection: Member Since

================================================================================

1.  equals
2.  is greater than
3.  is less than
4.  is not equal to
5.  is blank
6.  is not blank
7.  contains
8.  begins with
9.  ends with
10. does not contain
11. does not begin with
12. does not end with

--------------------------------------------------------------------------------
Type number, or use arrows, then press Return                        50K Avail.
```

Figure 7-5. **AppleWorks provides a dozen options to searching for data base records according to your criteria.**

With pfs:File, you can create on the screen a form that closely resembles almost any paper form you may already use, since you can place the item headings anywhere on the screen, even across more than one page. An entry can contain lengthy comments or even complete quotations you might want to use in the paper.

A special feature built into the *II*c version of the program can also save some keystrokes on repetitive entries. If you're quoting several sections from the same book, for example, you can store the author's name, title, and other bibliographic data (up to 40 characters) in a separate memory space. Then, when you need to type the same information on the next record, you can do it all with the press of one command. In computer jargon this feature is commonly called a *macro*.

To help you sort your research "note cards," you can create one item titled Keywords, which contains important subjects addressed in the note below it. Then, if you want to see all your records on a particular subject, you can sort your records by the keyword. At the same time, the program searches through every word of the note to find references to people, places, and things. Setting up a note-card format also lets you create fields for the books and articles from which the note comes. At the end of the project, sort your records by authors' last names, and have the computer print out the file (with pfs:Report) formatted as a bibliography at a few hundred words per minute.

INVENTORY MANAGEMENT

Another practical data-base application for the *IIc* is managing inventory. You might not think there are many household applications for inventory management. But an extensive hobby collection, like stamps, coins, or books, could probably use cataloging to track the items on hand and their values. If you make a lot of cassette tapes from your record albums to take with you wherever your Walkman goes, then you might find it worthwhile putting your record collection on computer, complete with the listing of tracks on each album and playing times. Then, when you're almost at the end of recording a tape side and need a short piece to fit in with the style of the rest of the tape, you can have the computer sort through the entire collection for a track by a particular artist lasting no more than a specified amount of time.

DISK MANAGEMENT

Similarly, your *IIc* is going to produce a significant collection of floppy disks. You'll have one or two dozen program disks in no time. And your storage disks will start piling up when you become productive on word-processing and data-base programs. You will find it beneficial to manage your disk library with the *IIc*. Several outside software developers offer programs especially designed for disk library management for the *IIc*. Using one will help you locate a program or storage disk you need by checking an on-screen or printed report of your holdings.

CHAPTER 8

The Apple *II*c as Telecommunicator

To most newcomers to personal computing, probably the most foreign concept of the entire realm of computer possibilities is telecommunications—linking your computer to another computer over the telephone line. It's hard to imagine why anyone would want to do such a thing, especially if you have no pressing business need that dictates such a hookup.

Believe it or not, however, most of us have communicated with computers over the telephone already. When you dial a telephone number in most North American cities, the numbers you press or dial activate a computer at the telephone switching office that sorts your request and makes the proper connection to the called party. More directly, if you subscribe to one of the discount long-distance services like Sprint or MCI, you might not realize that when you dial the local phone number and hear the answer tone at the other end, you are actually hooked up to a computer. By pressing the buttons corresponding to your access

code, you are telling the computer to let you into its system to make a long-distance call. Finally, if you use an automatic teller available in many banks around the country, you are actually using the teller as a remote terminal to manipulate information stored in a central bank computer elsewhere in the area. The remote terminal is connected to the main computer by a telephone line dedicated to computer information transfer.

In Chapter 6, we saw how the *II*c can be used to retrieve stock-market quotations over the telephone with the help of a telephone modem and communications software. In this chapter we'll see what else you can do with that hardware/software combination.

HOME BANKING

Until recently, the idea of performing banking chores from a home terminal was more like something out of Buck Rogers than real life. But a number of banks and other savings institutions around the country are improving on the concept of automatic tellers by allowing you to access information about your accounts twenty-four hours a day from a computer rigged with a modem. In fact, home computer banking, as implemented by a number of banks, gives you much more power over your finances than the automatic tellers—for everything except getting cash, of course.

For the *II*c, all you need is a modem and communications software like Apple Access *II* (Apple), although some banks may offer software for the *II*c specially designed to help you gain authorized access to that particular computer. Apple Access *II* not only turns the *II*c computer into a telecommunications terminal,

but also adds a level of communications intelligence to the machine and modem. For example, you can store on the disk both the phone number of the bank's computer and your personal access code or password. That way, when you're ready to do some banking, you start the computer with the Apple Access *II* disk in the drive, turn on the computer, and select the bank's service from a directory shown on the screen. The directory shows all the services for which you have previously stored telephone numbers, passwords, and communications parameters (for an explanation of the baud rate parameters, see the section on modems in Chapter 3). The computer automatically dials the call and logs onto the bank's computer for you. From then on, you follow instructions and select from screen menus presented by the bank's computer. If you want, you can store information displayed on the screen by directing the computer to copy the data to a disk. You can review the information later, when you're off-line.

Among the more common home computer banking services are those that you might already perform with an automated teller. One of the most productive is the ability to move funds among various accounts you have with the bank. If you need money in a checking account to cover checks you've just written, you can call up the bank's computer and instruct it to transfer the appropriate funds from an interest-bearing account to your regular checking account. If you need to keep a minimum balance in various accounts, you can keep tabs on your daily balance to see if you're dipping too low.

More sophisticated services include the ability to see which checks have cleared and when deposited funds

are available for your use. Many home banking services now also tie into local merchants and utilities so you can actually pay most of your bills on the computer, without having to write a single check. You prearrange with the bank as to which account you want to use to pay bills. Then you can transfer funds from that account to the payee. Since the transfer is instantaneous, you can wait to the last possible day before finance charges are added before you pay the bill—keeping the funds in an interest-bearing account in the meantime. You also don't have to worry about being assessed a late charge for the delay caused by a check being held up in the mail. You know instantly when you've paid a bill.

Home banking by computer is still in its infancy, but the few banks that have experimented with it are finding that their customers like the service. There is usually a small monthly charge for the service (the Bank of America in San Francisco, for example, charges $8 per month). But some industry sources predict that in the future, banks will discourage face-to-face banking with a combination of fewer open hours and perhaps higher charges for personal service than for the computerized service. In light of such predictions, it may be comforting to have the *//c* on hand as a banker whenever you need it.

RESEARCH

Apple Access *//* software can also turn your *//c* into a very powerful research tool for school and business by hooking your computer up to commercial data bases like Dialog (and its consumer version, Knowledge

Index, both from Palo Alto, CA) and, to a lesser extent, CompuServe (Columbus, OH) and The Source (MacLean, VA). In their most basic senses, these data bases resemble the service offered by the Dow Jones News/Retrieval service noted in Chapter 6. To gain access to them, you dial a local phone number which ties into the central computer system of the service. You make inquiries about various subjects carried on the service, and the information comes scrolling across your screen. Important data that will be of value to you later can be saved on disk—downloaded—as it arrives. Once you disconnect, you can print out information saved on disk (a printer is slower than the incoming data, causing you to stay on the line longer, raising your connect charges).

Dialog Information Services has the most extensive collection of computer-obtainable information you'll find anywhere. It is actually a collection of many data bases prepared by other information companies. The list is extensive and covers many different categories of particular interest to business and science or anyone looking for information in these areas. For example, you can search through dozens of magazines and national newspapers in a few minutes to find references to subjects you're interested in. The output from Dialog includes bibliographic references and article abstracts. A search through a Dialog data base is similar to a search through a data base created on AppleWorks, except that the collection on Dialog is much larger, sometimes in excess of one million records. If you like, you can order photocopied reprints of the complete articles while you're on-line with Dialog, but this service is rather costly for individuals. Corporate research departments, however, take ad-

vantage of this all the time. As an individual, you might prefer to do your primary sorting through magazine and newspaper indexes on-line, and then head for the local library to retrieve complete articles.

Here is a sampling of data bases selected from the nearly 200 available from Dialog:

Adtrack—an index of advertisements of 1/4 page or larger appearing in 148 consumer magazines.

America: History and Life—index of abstracts, citations, book reviews, and bibliography on the subject.

American Men and Women of Science—biographical index to 130,000 scientists in the U.S. and Canada.

Biosis Previews—index to over 4 million articles and papers in the life sciences.

Book Review Index—location guide to humanities-oriented book and periodical reviews in over 380 journals.

Books in Print—a computerized listing of the respected reference work found in libraries and on publishers' desks.

Bureau of Labor Statistics—four different reference data bases: consumer price index; employment, hours, and earnings; labor force; and producer price index.

Child Abuse and Neglect—guide to research and published work in the subject area.

Computer Database—abstracts from 530 periodicals and technical sessions, including product evaluations.

Economic Literature Index—index to articles and reviews from journals and papers.

Electronic Yellow Pages—a nationwide finder service in several industries: construction, financial services, manufacturers, professionals, retailers, services, and wholesalers.

Encyclopedia of Associations—information about trade, professional, labor, fraternal, and patriotic organizations.

Environmental Bibliography—index to articles covering ecology, land/water resources, nutrition, and health.

ERIC—collection of research reports, projects, and articles covering the educational profession.

GPO Monthly Catalog—a computerized version of the Government Printing Office's catalog of documents offered for sale.

Grants—listings of over 2,000 grant programs from the government, corporations, associations, and private foundations.

Harvard Business Review—complete text of articles since 1971.

International Software Database—listing of personal computer software currently available.

Magazine Index—index to general-interest articles in over 400 magazines.

Marquis Who's Who—biographies of 75,000 noteworthy Americans.

Medline—index to articles from over 3000 biomedical journals in 70 countries.

National Newspaper Index—index for entire contents of *Christian Science Monitor, Wall Street Journal, New York Times, Los Angeles Times*, and *Washington Post*.

Newsearch—daily index of thousands of stories from over 1700 newspapers and periodicals (later added

to National Newspaper Index, Magazine Index and others).

Peterson's College Database—information about two-year and four-year colleges in the U.S., territories, and Canada.

Philosopher's Index—index and abstracts from books and journals in philosophy.

Religion Index—index and abstracts from theological literature.

UPI News—full text of stories sent over the United Press International wire.

Dialog is not necessarily a consumer-oriented data base, and its rates tend to prove it. Connect charges vary from data base to data base, but on average fall in the $50 to $65 per hour range. Although you're only charged for the minutes you are on-line, unless you know how to organize a search, this can be a costly research device. Many professionals, however, find the time it saves in manual library research to be worth the high fees.

Some of Dialog's data bases that appeal to consumer and school research are available on a less expensive service, called Knowledge Index. For $35 per on-line hour, you can access any of more than a dozen selected data bases during evening and weekend hours. Of the Dialog data bases listed above, you can access Magazine Index, Newsearch, National Newspaper Index, Medline, ERIC, and Books in Print, among others, through Knowledge Index.

CompuServe and The Source are consumer-oriented on-line data bases, with connect charges in the under-$10 range during evenings and weekends. As research

tools, they offer some services different from Dialog and Knowledge Index. For example, CompuServe provides one of the few on-line encyclopedias available at a reasonable rate: Grolier's Academic American Encyclopedia. To pull out an article, you specify a keyword, and the central computer searches the encyclopedia for the article you need. You can download it as it comes off the data base, and print it out later as reference material when writing your paper.

The Source offers lower-cost access to the UPI data base than that available on Dialog. You can also search through Management Contents, a collection of article abstracts from important business magazines like *Business Week, Forbes,* and several dozen others.

ELECTRONIC MAIL

A growing application for telecommunicating computers like the Apple *II*c is electronic mail. Both CompuServe and The Source provide electronic mail services for their subscribers. There are also outside electronic mail services, like MCI Mail, just getting under way. Basically, you send messages to other computer users who are also subscribers to the same service.

To send a message via The Source's Source Mail, for example, you sign onto the service and type in the command indicating you want to send a message, plus the account number of the person to whom you're sending the message. Every subscriber has a unique account number (plus a password). Account numbers in electronic mail are identical to mailbox addresses in regular mail. You can't charge anything to an account

number (like a charge account) unless you also have the password—something you keep to yourself. Onscreen instructions guide you to fill in a blank about the subject of the message (which will appear on the heading when the recipient scans incoming mail) and to start typing your letter. With the *II*c, modem, and Apple Access *II* software, you can write your message before you log onto the system, and then send it at lightning speed from the disk (uploading), all of which cuts down on your connect charges.

SENDING DOCUMENTS

Perhaps a more specialized application of telecommunications on the *II*c is what many professional writers do all the time: send their stories and other documents to editors via telephone. With the easy transportability of the *II*c, it's likely that writers now using deskbound Apple *II* computers will add a *II*c and modem to their travel bags. Then all they need is a telephone to file their stories with editors back at the head office.

These same people will also be able to check in with a second Apple (or other computer) back home. By leaving the computer at home turned on and the modem in an auto-answer mode, it stands ready to receive any computer calling in to it. You can upload and download (send and receive) messages, documents, and programs from your homebound computer. Family members (or business associates, if the base computer is in the office) can leave messages for you that you can retrieve whenever you get a chance to call in. Likewise, you can leave instructions for those back home to follow the next day when they read what messages came in overnight.

GETTING HELP, MEETING FRIENDS

This kind of remote access to a computer—even a computer like the *//*c—is actually quite common. Hundreds of local hobbyists and computer clubs have set up computers acting as electronic bulletin boards. Access is usually free to any caller who has the modem and software, like Apple Access *//*, to check in, although some are available to members only. These boards are predominantly message centers, where, for instance, you can ask questions about your computer, hoping that someone has the experience or knowledge to help you out of a confusing spot with a piece of software or peripheral. Eventually, you will start communicating regularly with one or more people on these boards and build friendships.

Much the same kind of help is available on the national data bases, The Source and CompuServe. There is much activity on CompuServe's Special Interest Groups (called SIGs). There's a SIG devoted to Apple computers, on which you can leave messages directed to all comers to help you solve a problem or perhaps provide insight into an advanced aspect of the computer you're trying to learn. In this case, however, the friends you make could be from anywhere in North America.

You might even arrange to use CompuServe to communicate directly with new-found friends using the service's Citizens Band simulation, a popular feature. In this mode, your conversations appear on each other's screens as you type.

TRAVEL PLANNING

A favorite communications service of traveling executives is looking up flight schedules and placing travel reservations. CompuServe offers an on-line version of the Official Airline Guide (OAG), although at a slightly higher connect rate than its standard services. With the OAG, you can check not only regular flight routes, but also connecting flights to smaller towns, as well as fares. Then you can bring up a different service within CompuServe and actually place your reservations online with a travel agency. If there is enough time before your departure date, tickets are mailed to you.

Over on The Source, you can check out hotels and restaurants in the towns you're going to. The Source also has an air schedule service, prepared by a company other than OAG, but it often contains errors, whereas CompuServe's OAG is more reliable.

SHOPPING

The whole intent behind electronic shopping is to relieve us of devoting what precious spare time we have to the drudgery and crowd-fighting of shopping in stores. Comp-U-Store, a national shopping service on CompuServe, lets you shop and purchase brand-name merchandise while connected to the data base. Shopping categories range from stereo gear to automobiles, often at substantial savings (e.g., cars at $125 over cost). All merchandise is delivered to you, often drop-shipped by a manufacturer, distributor, or local dealer.

Electronic shopping is only now starting to make its way to the local level. It's possible that in the near

future, you will be able to order groceries and staple items by computer. In one setup, a store has professional stock pickers who take your order and assemble it for you to pick up almost on a drive-through basis. In another system, the store will deliver the goods right to your door.

CORPORATE COMPUTER ACCESS

Many large companies are opening up their computers to allow executives and field salespeople access to information via telephone. A manager can download sales performance data that will become the foundation of a spreadsheet calculation, forecasting sales for the balance of the year. Traveling salespeople can obtain customer information the night before a sales call to prepare a presentation on a new promotion. Orders can be entered practically from the customer's desk, with a scheduled shipping date supplied by the main computer.

All in all, telecommunications is putting more people in instant touch with more information. It is perhaps the most exciting aspect of personal computing, and something every *//c* owner should try.

CHAPTER 9

The Apple *II*c as an Artistic Tool

We're all familiar with computer graphics in one form or another. Arcade and home video games use computers to generate elaborate color pictures on the screen. What we see on the screen is often the result of painstaking programming efforts of trained graphics artists and experienced computer engineers. Knowing that, it seems hard to believe that a personal computer could be much of an artistic tool for someone who has trouble drawing a straight line with a ruler. But graphics software turns the Apple *II*c into a practical addition to the business and professional workplace. Used at home, computer art can be fun as entertainment, and can also unlock an amazing amount of talent hiding inside of you.

In this chapter, we'll be looking at two different kinds of graphics programs for the *II*c. One is definitely directed at businesslike uses—which include home budgeting and managing the numbers of an organization. The other, at first glance, is nothing more than a toy. But in a number of situations, it becomes a most

practical enhancement to the writer's art, not to mention that of a true graphics artist.

PRESENTATION GRAPHICS

When it comes to making a business presentation—either a written report or a live show before a group of people—there is nothing more boring to look at than page after page of numbers. To interpret the figures correctly takes concentration and time. Many executives prefer to pass over the details and just get to the bottom line. If you can convert those figures into meaningful charts and graphs, the figures come to life. It is then much easier to spot trends and put the figures in perspective.

In some companies, managers have access to art departments that can take a rough sketch of a chart and turn it into something worthy of a presidential address. But not everyone has that luxury of staff or time. The problem is, then, how to convert the figures from a spreadsheet into colorful charts and graphs.

That's where the *//c* comes in. With programs like pfs: Graph (Software Publishing), the computer does most of the work in a flash. If you don't like what you see, or if a figure changes at the last minute, simply select a different option from an on-screen menu. With no questions asked or complaining, the *//c* redraws the entire chart.

Presentation graphics programs are designed to reduce the amount of work you do to a minimum. But there are a few things a program must know before it can

draw your graphs. The first thing it needs is the figures. Since in most charts and graphs you'll want to plot one factor against another (e.g., sales against time, overtime hours worked against the order backlog), you have to supply figures for both an x-axis (vertical) and a y-axis (horizontal). Most graphics programs let you input this data either directly from the keyboard or loaded in from a spreadsheet. VisiCalc, the pioneering spreadsheet program, started a method of saving information so other programs would be able to use it. The method is called Data Interchange Format, or DIF for short. Graphics programs, therefore, usually accept DIF files as valid input for the x- and y-axis figures.

Next, you have to tell the program what the labels are for each axis. If one axis is raw numbers (e.g., units sold), then you usually need to specify either a maximum number or a range of numbers. The scale of the axis and the increments are calculated automatically for maximum legibility. Finally, you can place a legend on the graph that informs a reader what the bars, graphs, or pie chart segments stand for. This is especially useful if you have more than one graph superimposed on another. For example, you might use the same vertical bar graph to show your sales and net profit for each month. With dollar units going up the vertical axis, and months shown across the bottom axis, you can place two bars side by side above each month's name. The graphics software understands that you want each graph line or bar to stand apart from the others, so it automatically creates a different area or line pattern (or different color in some programs when used with a color monitor) to differentiate the data. One bar, a solid black perhaps, represents sales, while a shaded bar indicates profits. Below the

graph are two small boxes with each pattern and legends for each: Gross Sales and Net Profit. Superimposing multiple plottings in this way on a single bar chart or graph can be particularly illuminating compared to the standard spreadsheet presentation.

Once you have created the chart or graph to your satisfaction on the screen, you have three major ways of making it presentable to your colleagues. If the program produces colorful video charts, you might consider bringing the *II*c and color monitor to the meeting and letting it produce the graphs in living color. With each chart stored as a separate document, you can line up their names so you can retrieve each one from the disk in succession with a minimum of keystrokes. If the group is not particularly attuned to personal computers, you may come off as a real whiz kid with computers—immediately intimidating the whole group. They're bound to be impressed with the graphs, even if the numbers don't look so hot.

For printed reports that are to get photocopied and circulated around the company, you'll probably just want to print out the graphs on an Imagewriter printer. A black-and-white printout may not have the same splash as a color video presentation, but at least the printer can capture the fine points of different patterns.

Overhead transparencies are a common information medium in corporate gatherings these days. Not only can you convert your graphs and charts to transparencies, but to color transparencies as well. With the addition of a color plotter and special narrow-tipped pens designed for plotting on transparency film, the *II*c and graphics software can turn out colorful, profes-

sional-looking foils in a matter of minutes. When everyone else is showing black-printed transparencies of numbers, your colorful graphics presentation will leap off the screen. A color plotter can also draw your graphs on plain paper in case you need a couple of specially colored hard copies of a presentation for a special prospect or the boss.

Presentation graphics software is getting much attention these days, as more executives don't mind tackling the job since they've already mastered spreadsheets and data bases. You're bound to see further advances in presentation graphics software, including programs that have libraries of preset formats from which you can readily choose.

TEXT ILLUSTRATIONS

One of the most revolutionary pieces of software ever to grace a personal computer is a program called MacPaint designed for the Apple Macintosh computer by Bill Atkinson. It places a remarkable toolkit of computer graphics features in the hands of anyone who can operate the Macintosh's mouse pointing device. And yet, as amazing as that program is, an even more fantastic program awaits every *II*c owner: MousePaint. It's not that MousePaint does more than MacPaint—it clearly does not. But that so many of MacPaint's features have been convincingly translated to a computer with less graphics resolution and smaller memory handling ability is a programming triumph for Bill Budge, who adapted the program's elements from MacPaint.

As you might guess from the title of this program,

MousePaint uses the AppleMouse as the main means of communication from user to computer. The program is packaged with the AppleMouse, and is one fantastic bonus for buying a peripheral. Figure 9-1 shows a MousePaint screen. Across the top is a menu bar. When you position the screen pointer on one of the menu words and press the mouse's button, a list of related commands appears below the menu word. Moving the pointer down the menu (called "dragging") causes each item it passes over to be highlighted. When you point to the command you wish to execute, you release the button, and it's done. The File menu contains commands pertaining to retrieving and storing pictures on the disk. The Edit menu contains many commands for cutting, copying, pasting, and some outstanding tools we'll describe later. In the Aids menu are several more useful tools that help you create on both a micro and macro scale, as we'll see in a moment. And the Fonts menu lets you select from five different type fonts for any text you may want to include in the picture.

Along the left side of the screen are a selection of *icons* that represent ways to put images on the electronic paper. Each icon stands for a different kind of operation. Selecting one is as simple as moving the pointer to it with the mouse and pressing the button (called "clicking"). The hand icon is what you choose to move the screen window around the virtual page with which you have to work. The virtual page is about twice the size of the window shown on the screen. You use the hand icon just as if the hand were your own. When you select that icon, the pointer turns into a replica of the hand. You place it anywhere on the "paper" and drag it with the mouse in the direction

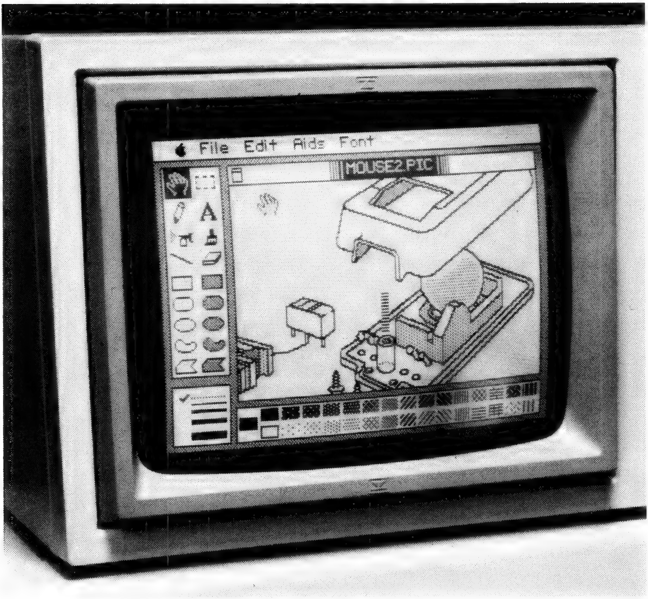


Figure 9-1. **MousePaint for the Apple //c replicates many of the features of MacPaint for the Apple Macintosh computer.**

you want to push the "paper" to see another segment of the virtual page.

The pencil, spray gun, and paintbrush are ways you can draw on the paper in response to mouse movement. The pencil produces a narrow line; the spray gun produces a dot pattern that gets thicker every time you go over the same spot; and the paintbrush produces one of sixteen different strokes depending on the brush setting you select from the Aids menu. To help you draw straight lines, you have the straight-line icon (the diagonal). After you select that icon, you move the pointer on the paper where you want the line

to begin, and click the mouse. As you move the mouse around the page, it draws a straight line between it and the original point. The thickness of the line is determined by the line icon you have checked at the lower left corner of the screen. Clicking the mouse a second time plants the second end of the line. Next to the straight-line icon is an icon for an eraser. Click this icon, and the screen pointer looks and behaves like a chalkboard eraser.

The ten shapes in the bottom of this box are special drawing aids similar to those found on very expensive graphics workstations. You can draw perfect rectangles, squares, circles, and rounded squares with some of them. Others let you create free-form shapes with curved lines or straight lines, as in a trapezoid. You can make all these shapes either in outline form or filled in with any one of the 30 patterns or colors selectable from the palette displayed across the bottom of the screen. Whatever pattern you select shows in the leftmost box of the palette. So if you draw a circle with a diagonal pattern selected, the circle is automatically filled with that pattern. You can create some dramatic contrasts of shape and texture by overlapping shapes filled with different patterns.

To add text, like labels, to a picture, you click the A icon, which indicates a text mode. The screen pointer becomes a text insertion pointer (identical to the one on the Macintosh). Wherever on the screen you click the mouse, the characters you type will appear there. If you don't get something placed just right (either text or a picture), you can move it around freely. All you have to do is click the Select icon next to the Hand icon. With this special pointer, you drag a rectangle

(click once to plant one corner, then drag the mouse to enlarge the area until you click again to plant the opposite corner) around any area on the screen that needs moving. Then, with the pointer located within the "magic rectangle," you move the mouse to position the contents precisely where you want them.

A special feature of the Aids menu is a function called Fatbits—a favorite of every MacPaint user as well. With Fatbits, the screen becomes a magnifying lens, so you can see individual pixels as big as life. Moving the page around with the Hand icon, you can bring into view any area that doesn't look quite right to you, and fix it up, pixel by pixel. You see both the giant pixels and the regular-sized representation of the drawing in the upper left corner of the screen. With the pencil icon, you can erase or fill in one pixel at a time to fine-tune the drawing. With the help of the regular-sized picture, you can experiment to your heart's content, creating the kind of detail that is impossible to capture when working in regular size only. This was the micro scale of working with pictures I mentioned earlier.

On the macro scale, however, another Aids menu feature is called Show Page. When you click this menu item, a special window appears on the screen to show you how the overall page looks. You can see if everything is centered properly without having to guess from looking at only a portion of the page.

Two interesting commands from the Edit menu are the Flip commands, one for horizontal, one for vertical flipping. These commands let you create mirror images of anything you put on the screen. As with moving an

image, you first place a magic select rectangle around the image you want to flip. Then invoke the command from the pull-down menu with the mouse.

Finally, a very useful editing tool lets you make multiple copies of images. If you want to place a certain shape in several locations on the screen, you can copy it and then paste it anywhere you want. The Paste command even turns the screen pointer into a pointing finger to show you where the image will be pasted the next time you click the mouse button.

Despite all these functions and the fine graphics detail available on this program, you probably still want to know where you can use such a program. Since you can print the pictures you create with MousePaint on the Imagewriter, you may find it useful to spark up a business presentation, article, or newsletter with some freshly drawn graphics. Writers who use word processing for their words should not forget that many times an illustration can ease the task of describing a complex subject. Use MousePaint as a picture processor for illustrations the same way you use a word processor for the text. If you use the //c as a professional writing tool, many editors and publishers pay extra if you supply the illustrations, instead of having their costly art departments do it all by hand. The output from MousePaint and the Imagewriter printer is fine for many publications.

MousePaint will find its way into more applications than you think. Homemade party invitations and maps are naturals. Eye-catching bulletin board notices and posters take on a professional tone when created with

the help of MousePaint. Eventually, you will find all kinds of ways to use the program, from rearranging the living-room furniture by drawing a floor plan to creating personalized stationery. A program like this tends to spark the imagination much more than any other kind of applications program.

CHAPTER 10

The Apple IIc as an Educational Tool

For years, the primary application of a computer in the home has been for entertainment—playing video games. But a new trend is emerging. Today, the predominant rationale home computer buyers cite for bringing a computer into the home is to expose their school-age children to the educational value of a personal computer. One reason for this shift is that today's home computer buyer is less of a computer hobbyist enamored by controlling the destruction of screen aliens, and is more concerned with using the computer for some practical end.

Many parents maintain a belief that, if their children haven't acquired computer skills by the time they graduate from high school, they'll have difficulty getting a job in this increasingly computerized society. Mistakenly, these frightened parents rush out and buy the first computer they see and plant the kid in front of it, secretly hoping that in a week's time the child will be able to reprogram a Defense Department computer.

What most parents fail to understand is that bringing a computer into the home for educational purposes is a big responsibility.

The Apple *//c* lightens that load somewhat because there are more educational programs available for Apple computers than for any other personal computer. Therefore, you won't have to dig through heaps of game programs at the store for an occasional educational program, something you have to do with most home computers.

The Apple *//* family of computers also happens to be by far the most popular computer among educators. A vast majority of student computers in schools today are Apples. While it is certainly not essential to have at home the same computer that your child uses in school, in some teaching environments it is a help. For example, if the child is learning a programming language on the school computer, he or she can bring home the programs to show parents or to work on over the weekend when there is less time pressure.

This last point—the time pressures in the schools—is a formidable reason for having a computer at home for children interested in using the computer for learning purposes. In most classrooms, there are too few computers available to give every student enough time to explore the intricacies of some learning programs and programming languages. Consequently, most of the programs used in schools are of the drill and practice kind, in which the computer is being used for little more than an electronic flash-card generator that operates at a student's own pace. But there are many more interesting learning programs designed for the *//c* that

are well suited for use in the leisurely environment of the home.

FOR ALL AGES

Preschoolers and elementary-age children need supervision with educational software, at least in the beginning. This means that for your children to get interested in using a computer for learning, you have to take an interest, as well. It will mean sitting in front of the computer with your child and encouraging participation. Often, you will have to demonstrate how things are done, since an on-screen prompt for an answer won't mean a thing to a three-year-old who does not yet know how to read. There will be times, though, when your child will discover how to use the program before you've had a chance to finish reading the manual.

Educational software is not only for the littlest kids in the household. As the shelves become crowded with basic arithmetic and grammar drills, educational software designers are building products that appeal to older children as they tackle high school subjects in science, math, and the humanities. Programs like Mastering the SAT (CBS Software) even help prepare high school students for the college entrance exams by providing practice in the kinds of skills these tests look for. The program pretests each user to determine the current skill level, and then tailors an instructional agenda around the areas that need the most work. Not ignoring the fact that the real exams are taken with a pencil and paper, two simulated practice exams are included in a companion workbook included with the program.

The *II*c can teach the whole family useful skills. With programs like MasterType (Scarborough Systems) and Kriya System's new Typing Tutor III with the arcade-style Letter Invaders (Simon & Schuster), the *II*c helps anyone overcome the fear of typing on the typewriter keyboard. Both programs combine instruction with a challenging entertainment environment, in which you improve your game scores by gradually improving your typing skills. MasterType also has lessons for learning the *II*c's Dvorak keyboard.

SOFTWARE CATEGORIES

Home educational software for the *II*c falls into five major categories: drill and practice; discovery; simulations; productivity; and computer languages. I'll cover the first four in this chapter, and reserve computer languages for the chapter on programming with the *II*c.

Drill and practice programs are by far the most prevalent. Many of the programs included in Apple's Education Classics software package are of this kind. They range from simple programs that duplicate arithmetic flash cards on the screen to some that really make you use your noodle. An example of the latter is called Darts, a program that teaches fractions and interpolation. The program produces a vertical line on the screen labeled with a number at each end, and sometimes numbered intervals in between, like markings on a ruler. For each round, the computer places three balloons at random along the ruler. The player's job is to determine at what measurement along the line the three balloons are located, and type in that number (integer, numerator, and denominator). The computer sends a dart toward the ruler at the typed-in measure-

ment. If the dart hits the balloon, it pops. At easier levels, the balloons are big and cover a wider area, making it easier to be slightly off target and still pop them. In this way, kids learn about fractions and interpolation.

There has been much exciting development in the discovery software area. A discovery program offers a more subliminal learning process, because instead of answering round after round of problems in a drill and practice fashion, a child becomes engrossed with the action on the screen, building knowledge from what has already happened. Spinnaker Software has established an impressive record of producing quality discovery software (among other kinds of educational products as well). A fascinating new Spinnaker software product for younger children is called Grandma's House. The screen becomes a make-believe world through which the child, in the guise of one of thirty different selectable characters, explores places like a beach, a park, a store, and a garden. Each is filled with objects associated with the surroundings. As if playing with a huge, well-equipped dollhouse, the child finds many different objects and brings them back to Grandma's house. The objects can be arranged throughout the house, where Granny has hidden some surprises for the lucky one who finds them. In the meantime, the child is learning a number of fundamentals about using a computer mouse or joystick to control movement on the screen.

On an even more fundamental learning level, programs like Stickybear Shapes (Weekly Reader Family Software/Xerox Education Publications), for ages three to six, teach very basic concepts. In this pro-

gram the child can use the AppleMouse (in the *//c* version of the program) or keyboard to pick out shapes (circles, triangles, squares, rectangles) in animated computer-generated cartoons featuring the Stickybear character. The program comes with a parent's guide, as well as colorful supplementary material to appeal to the child.

The third category of educational software is simulation. These activities are fun for both children and adults, since they often appear to be using artificial intelligence to guide the player through simulations of business, scientific, or other activities. On one of the Apple Education Classics disks is a program called Lemonade (it is also included on one of the disks, Apple at Play, included with the *//c*). In this simulation of supply and demand, a single player or up to four competing players run individual lemonade stands. Each player knows the other is running a lemonade stand. They know their cost of materials and must determine at the beginning of each day how many glasses of lemonade to make (based on the weather forecast and how aggressive they will be in advertising and pricing), how many advertising signs to buy, and how much to charge for each glass. At the end of each day, the computer presents a summary of the number of glasses sold for the day and calculates the profit or loss for each player. This simple activity teaches subtle lessons in marketing, economics, and business. Kids can play it, but adults will learn something from it, too.

Another simulation, for ages nine to adult, called Rocky's Boots (Learning Company) is a remarkable experience for learning the fundamentals of logic and

even a little about elementary electronic circuit design. The screen becomes a work area in which you assemble logic diagrams (using the same symbols electronic designers use to assemble logic circuits) in order to make some of the components behave in such a way that you acquire high scores. The program contains 40 different activities, and is the kind of program that you can spend hours with without realizing it.

The fifth educational category consists of productivity programs that can be learning exercises or real productivity tools. An example of a learning and imagination-building exercise is a program called Story Teller, one of two programs packaged in Fact and Fiction Toolkit (Scholastic Software). Using the optional AppleMouse on the //c, a child can draw colorful story screens and add text lines in several different type styles to make up panels for stories on the computer. If a printer is available, the child can print out the panels to share with friends. There is also a library of double-high-resolution graphics shapes already stored on the disk, any of which can be selected and added to a screen picture.

The other program in this package, called Secret Filer, is a practical productivity tool as well as a learning exercise. With this program, a child creates an electronic card file (a mini data base) with up to five different one-line items per card—perhaps a catalog of friends and their addresses. The AppleMouse is used to select menu options throughout the program. At the same time the child is learning about computer filing, he or she also learns the importance of organizing information.

Word processing is another practical productivity tool that older students should take advantage of. Presenting a carefully edited school paper to the teacher printed out with nary a smudge or erasure will likely contribute to a better grade. See Chapter 5 for more on word processing for schoolwork.

COMPUTING FOR TEACHERS

Teachers are using computers today for a lot more than just supplements to classroom work. Data-base and spreadsheet software can be readily adapted to classroom paperwork of recording grades and attendance.

The spreadsheet sampler provided with the AppleWorks program shows how a teacher can use the computer to both record individual assignment grades and then automatically calculate the student's average score for the term. At the same time, the teacher can perform analyses of the grades for the entire class to assist in self-evaluation of tests and assignments. For example, if the class average for a particular quiz drops way below that of similar quizzes, perhaps the material has not been grasped by the class in sufficient detail, and a refresher is in order. Likewise if a graphical trend analysis of weekly homework grades reveals a strong upward swing, perhaps it's time to quicken the pace of learning, since the class doesn't seem to have difficulty with the material so far. Without the ease of the computer analysis, the grades might have been just numbers in a grade book.

Teachers don't have to design their own data bases and spreadsheet templates, however. Several software producers offer specialized grading and attendance

programs already tailored to the classroom environment.

EVALUATING EDUCATIONAL SOFTWARE

Making a choice from the hundreds of educational software programs on the market is no easy task, nor should it be considered lightly. There is, frankly, a lot of bad software out there. Fortunately, there are some guidelines to follow when shopping for educational software to be used at home.

First of all, unless you have seen the product on someone else's computer or have read a glowing review about it in a computer magazine, avoid buying an educational program without getting a live demonstration. With the //c, this is much easier to do than with the low-cost home computers, because //c dealers are trained to take the time to help you select the right software.

Before you run the program, skim through the manual. There should be quick start-up instructions in the front, showing you how to load the program and to begin using it. Next, the instructions for using the program should be straightforward and relatively easy to grasp even while standing in a computer store.

Being working with the program and be on the lookout for some key characteristics. First, if the program is a drill and practice type, it should offer positive reinforcement when you make a mistake. It should encourage you to try again, not just give you an impersonal electronic raspberry. The program should be graphically interesting, especially if it is targeted for

young children. Color graphics and lots of animation make computer programs much more inviting to kids, who will look at the computer as a cartoon show.

The program should offer increasingly difficult levels of challenge and variety in the types of activities. The last thing you want to happen is to have your child master a program in one evening. That's instant death to an educational program. But if a program offers depth by making available a number of user-controlled variables (like letting the child assume the identities of different characters in a storytelling program), then there will be much more incentive for the child to want to use that program again.

Ideally, a program with several challenge levels should keep track of how far along you've progressed, so that the next time you use it, you will be reminded of how many lessons or activities you've covered, and where you should pick up. A program like this can even keep track of several different users, since every time you load the program, it asks for the name of the user.

The program should be forgiving. By this, I mean that if you make a mistake by pressing the wrong key—the Delete key instead of a number key, for instance—the program will recognize it as a wrong key and remind you to press only the right kind of key. Poorly designed programs can easily "crash" if you throw in a monkey wrench, and any score you've acquired will disappear as the program freezes or automatically quits. A forgiving program has been programmed to expect the most illogical input and will know how to react to it without making you feel like a fool.

Other things to be on the watch for are simple on-screen instructions, so you don't have to run to the manual every time you want to make a move. There should always be some option or instruction on the screen so you are never at a loss for what to do next, even if it is a command that pauses the program while a help menu appears. In educational software other than those programs that encourage creative writing skills, the typing should be held to a minimum, with shortcuts like pressing Y for a Yes response. A program should also let you select any segment of the entire program at any time during the session, thereby giving you the chance to redo an activity with which you had some difficulty.

Overall, be selective in the software you bring home for your children. You'll be spending a lot of money for each program ($20–$50), so make sure the program will sustain interest and be a valid learning tool at the same time.

CHAPTER 11

The Apple *IIc* as Entertainment

If you go into a computer store and look over the software racks for Apple *II* computers, you will be impressed by the apparent majority of entertainment software. Part of the preponderance of games in the stores is due to the popularity of the category—dealers stock the products that sell the most. But it is no illusion. Game software for the Apple *II* family of computers is plentiful. And why not? A personal computer shouldn't be all work and no play. As long as the machine has the capability of producing color graphics and sound, we may as well enjoy entertaining programs that offer a diversion from the serious stuff.

ARCADE HITS

Entertainment software takes on a number of different forms. The most obvious to newcomers is the translation of arcade video games. Typically, these are full-color, high-action, hand-eye coordination games adapted from twenty-five-cents-a-play machines. Hard-core arcade game players usually aren't as excited about the home versions of their favorite games,

because the computer can't fully re-create the action of the professional model. One reason is that the home versions are designed to use a color television set as a display device, while the arcade versions use very-high-resolution computer monitors and have the high-resolution graphics computers behind them to generate stunning displays. The result, then, is a less graphically pleasing game for the dyed-in-the-wool game player. But for the person who only infrequently or never drops quarters into the arcade machines, home versions are often the most exciting things playing on the computer.

Virtually every popular arcade hit has been translated for the Apple *//* family of computers. Zaxxon (Datasoft) was one of the most popular arcade hits of 1983 and set a new standard for arcade game graphics resolution. You command a space fighter as it flies over three-dimensional-looking, three-quarter-view space battlefields, with various surface-to-air missiles firing out of silos in front of you, and other enemy targets on the ground. Control of your fighter by joystick gives you control over altitude and lateral movement, all at constant speed. Occasional barriers and force fields give you only a sliver of open air to sneak through, or else you lose your ship. If you survive the battlefield and free space battle with opposing fighters, you then have to outguess the huge Zaxxon armed robot by anticipating the multiple missiles it fires at you.

TEXT ADVENTURES

Another popular kind of entertainment software for the *//*c is called the text adventure. These programs don't rely on colorful graphics images at all. Instead,

paragraphs of text try to produce images in the minds of the players. The stories scrolling on the screen take the user into other worlds or unique situations full of puzzles, mysteries, and clues, which must be put together for a final solution or for a better score than last time (points are awarded for making intermediate solutions). While the arcade style of game is usually a joystick bender, these games are mind benders. They draw you deeply into a labyrinth of scenarios, which you can either try to remember or make notes about for yourself. The solutions are not something you can figure out with the first play. A well-written text adventure may never be completely solved. You quite often find new clues or treasures every time you play or try a new approach.

What makes a text adventure game so intriguing is that not only does the program make statements (for instance, about what the underground cavern you just fell into looks like), but it responds to your keyboard input with an air of artificial intelligence. The latest programs, like the popular Zork series from Infocom, can understand instructions or questions typed in everyday English. You give instructions about your next move (e.g., "Wave the lantern"), and the computer branches to another part in the program based on your input. On most adventure games, since you usually invest a great deal of time unraveling the mysteries or plotting the layouts of the magical kingdoms, you can store your location on disk and return to the exact spot the next time you have time to sit and play. In addition to the entertainment value, it's worth playing with one of these programs just to see how a computer can be programmed to take on the semblance of another intelligence.

STRATEGY GAMES

More traditional strategy games are also available for the *II*c. Chess, of course, is a popular computer game, but the program How About a Nice Game of Chess! (Odesta) goes farther in presenting a self-contained program. First of all, there are six lessons to teach the fundamentals of the game to beginners. Once you start playing the game, if you find that you made a serious error and that the game is about to be lost, you can take back any number of moves and repair the damage. Upon request, the program offers advice about your next move. As a special enhancement for the *II*c and AppleMouse (or *II*e with AppleMouse adapter), you can play the game without touching the keyboard, using only the mouse to make all your moves and other commands.

SIMULATIONS

Sometimes education and entertainment overlap. That should be obvious from an educational program like Typing Tutor III, which has as one of its activities a game called Letter Invaders. But simulation activities also bridge the two categories. A program like Lemonade, although educational in intent, can be easily disguised as solely entertainment. Other simulations, notably aircraft flight simulators, might not teach you everything you need to know to perform the real-life task, but they do help you develop other skills. A flight simulator is a good lesson in managing more than one thing at a time—keeping the aircraft level, adjusting your course, watching the dials, etc.

Don't scoff at entertainment software. Games provide as important a balance to your *II*c software collection

as Monty Python does to public television. And by carefully choosing entertainment software for your *//*c, you can make sure that the time you spend with the computer will be productive—if not for turning out the Great American Novel, at least for helping you cope with juggling several things at once on the job.

*//*c COMPATIBILITY

Since practically all game software on disks is copy protected and since some copy-protection schemes involve highly technical aspects of a particular model of Apple *//*, some software designed for the *//*e might not work on the *//*c right away. But most Apple *//* software suppliers have had early warning of the changes made to the *//*c system and should have their software adapted to the machine by now. To make sure the game software (or any software, for that matter) will work with the *//*c, check the software packaging. It should specifically state on the package that the program works on the Apple *//*c (it will probably also include a list of the other models, as well).

CHAPTER 12

Programming on the *II*c

If you read my discussion in the Introduction about computer literacy, you already know that computer programming is not one of its prerequisites. Just the same, learning to program the *II*c can be a rewarding experience, even though it takes time, patience, and practice, just like learning any foreign language.

WHY LEARN PROGRAMMING?

If basic computer literacy can be compared to a high school education, then learning to program is like going to college. With programming knowledge, you begin to understand much more fully how computers "think." You can gain important insight into the inner workings of programs in general, even those written in a language you don't know. The "insides" of the computer will become less of a mystery to you, and you will see how simpleminded the machines really are. You'll see that they follow very specific rules and aren't any more intelligent than you make them. How many people do you know who would add one plus one all day without thinking the task to be stupid? The

computer, you'll learn, will continue performing the most ludicrous chores you tell it to. So it's not so smart after all.

Learning a programming language teaches you the importance of precision. Because of the computer's inability to understand anything but its own vocabulary according to specific rules of syntax, it will be quick to alert you that you have made a mistake in trying to give it an instruction. With this burden of precision, you improve problem-solving skills by learning to break up a big job into digestible pieces, and attack each piece one at a time.

Finally, there is still a thrill that experienced home computer programmers get whenever a program is finished and runs without errors. It may be a simple program, or something that has taken weeks of effort. Either way, the feeling of accomplishment at having mastered the machine is one that few other hobbies provide.

RECOMMENDING A LANGUAGE

You will be amazed at the number of similarities between human languages and computer languages. Aside from the more common parallels, like syntax rules and dialects, you also find proponents of each language reflecting the fervor and pride of a nationalistic campaign. One programmer's dream language is another's nightmare.

Thus, it becomes a dangerous assignment to recommend computer languages to newcomers. A programmer who designs local area network communications software has completely different language needs than

does a concerned parent who wants to design a game activity that teaches her children how to tell time and dial a home phone number.

INTERPRETERS, COMPILERS, ASSEMBLERS

Adding to the confusion for newcomers to programming languages is an array of terminology that is enough to intimidate the most persistent seeker of knowledge. But an understanding of the three most used terms—interpreter, compiler, and assembler—will help you in selecting your language(s).

Perhaps the best way to comprehend these terms is to take a closer look at what a language is doing inside a computer. At the most elemental level of the computer, you have the microprocessor—the chip that runs the whole show. By itself, the microprocessor is not too bright. In fact, the only way it can communicate with the rest of the computer is through an extremely rudimentary language consisting of two "words." In truth, the words are not words as we know them, but rather the 1s and 0s we learned about earlier.

Microprocessor designers build into each type of chip a unique vocabulary consisting of commands or instructions corresponding to very specific patterns of 1s and 0s. Thus, combinations of these pulses are the only instructions a chip understands. This is called "machine language"—meaning the language of the microprocessor and associated circuits.

Writing machine language programs is probably the most tedious job a programmer can have. Instructions

for even the simplest operation require more patience than most programmers can bear. After several pages of 1s and 0s everything begins to look alike and the chance for errors increases considerably. Fortunately, we have the benefit of experts who have devised other languages which are more human-oriented tools. These computer languages, as we know them, are designed to help us formulate a function or operation in our minds in terms we can better understand, and to then use the power of the computer to translate that operation into machine language.

Currently there are two popular ways of translating computer languages (which humans understand) into a machine language (which computers understand). The fundamental difference between the two ways is whether the translation takes place every time the program runs or just once after the program is written.

The first way is to interpret the words from a computer language into machine language every time the program is run. Let's use an analogy to explain what I mean.

Suppose a visiting head of state from Upper Cupertinia wishes to deliver a speech as he tours our country. He doesn't know our language, so he must have his speech translated to his audiences as he speaks. The advantage of having the speech translated this way is that as he tours he can make changes or additions to his speech before speaking to each audience. The major difficulty is that every speech takes a long time because he must pause periodically as the translator delivers the message in English. This is what using a computer language *interpreter* is like. When we run

the program, the computer sends every instruction in our program through an interpreter on its way to the microprocessor.

The primary advantage of using this kind of an interpreted language is that the programmer can make corrections or changes to the program very easily and then test the new program by immediately running it through the interpreter. The main disadvantage is that the interpretation of each program line takes time, and the program runs rather slowly.

Now imagine this: if our visiting dignitary has his speech translated only once, before his trip, the translator could simply read the entire speech at every gathering as if delivered by the guest of honor himself. Everyone in the audience would get home sooner. If the dignitary wanted to change the speech for Des Moines, however, he would have to take time out to write a new speech and have the translator prepare an updated version.

And so it is with a computer language *compiler.* After you write your program (with a word-processor-like editor), you put your program through another program, called a compiler, which converts the program into machine language. Thereafter, every time you run the program, you automatically supply the computer with the ready-to-run machine-language version that runs faster than through an interpreter. The disadvantage, however, is that making changes to the program means that you have to compile the entire program all over again.

Finally, we come to the term *assembler.* It is actually nothing more than a specially designed compiler for a

language called Assembly Language. Used predominantly by professional programmers (and very dedicated hobbyists), Assembly Language is about the closest you can get to working directly with the microprocessor without the tedium of 1 and 0 machine language.

With all this in mind, I will discuss each of the major languages you're likely to encounter in your search for a computing tongue. Most of our time will be spent talking about BASIC and Logo, since you will be exposed to these two languages on the disks packed with the *//c*. Throughout the discussion, I will direct my comments to those who are making their first foray into computer programming. If, on the other hand, you are an experienced programmer, you probably already have a good idea as to which language you want to pursue, or at least you have colleagues who can guide you in your professional language software search.

BASIC

Built into the Apple's ROM chip is a programming language called BASIC, which stands for Beginners All-Purpose Symbolic Instruction Code. Virtually every computer with a built-in BASIC language has its own dialect (each with differences in the command words and the way the words must be organized), and the Apple *//c* is no different. Its BASIC dialect is called Applesoft BASIC, an Apple version of a popular dialect called Microsoft BASIC (the founders of Microsoft Corp. invented BASIC for personal computers).

Learning the essentials of BASIC is made relatively easy because most of the words in its vocabulary are

English words or abbreviations thereof. The Plot command plots a line on the screen between two points you specify with the command. The Print command "prints" a copy of a word or a value on the display screen. PR#1, followed by a word or number, "prints" that information to the output port labeled #1, which on the *II*c corresponds to the printer connector. In all, there are 119 words in the Applesoft BASIC language that do everything from clearing the screen to moving information on and off a floppy disk.

The vocabulary of BASIC is divided into two groups, which behave somewhat differently. One group, *commands,* function the minute you type them on the BASIC screen and press Return. Commands often help you manage a BASIC program stored in memory. For example, the List command displays on the screen every line of a program currently in memory. The Save command instructs the computer to store a program in memory onto a disk for safekeeping.

Other words in the BASIC vocabulary are known as *statements*. These are placed inside a program, and are executed not when you type them, but rather when you instruct the computer to run the program with the Run command.

Some words in the BASIC language function as both statements and commands. For example, you can switch from text mode to high-resolution graphics mode manually by typing the HRG command. If you include the same word in a program as a statement, the computer switches to graphics mode when the word is encountered.

What BASIC Looks Like

BASIC programs follow a specified sequence of events, determined by a line number that you type at the beginning of every program statement (although more than one statement can appear in one line, provided they are separated by colons). Here's what a simple Applesoft BASIC program looks like:

```
10 INPUT "What is your name? ";X$
20 HOME
30 PRINT "Hello, ";x$;"!"
```

Line 10 uses the Input statement to print on the screen the question between quotation marks. Whatever you type in next will be assigned to the variable X$. The dollar sign signifies that the characters you type in are to be taken literally as a word ("string" in BASIC jargon), not as a number for subsequent math operations (if the variable was simply X and you tried typing your name, the computer asks you to reenter, since the plain X variable accepts numbers only). In line 20, the Home statement clears the screen and brings the cursor up to the upper left-hand corner of the screen, where it is ready for the Print statement in line 30. That line displays the word Hello, plus the name assigned to X$ in line 10, plus an exclamation mark.

Applesoft BASIC is certainly not the most powerful programming language available for the //c. But it is built into the computer, so you have no further expense other than the Applesoft tutorial and reference manuals available from Apple dealers. Computer and book stores also have many tutorials written by outside authors to help you learn and master the language. Moreover, if you subscribe to one or more

Apple-specific magazines (which I strongly recommend), you will frequently find listings in them of free programs you can type in from their pages. Those listings, by the way, are often a good source for learning new techniques that tutorial manuals may not reveal.

Applesoft BASIC in the *II*c ROM is an interpreted version of the BASIC language. Therefore, programs you write with it are easy to test and edit, but tend to run much more slowly than probably any language available for the *II*c. Yet for many applications, the speed of interpreted Applesoft BASIC on the *II*c is plenty fast.

The Applesoft BASIC Compiler, an optional software package from Microsoft, lets you compile BASIC programs developed and tested with the interpreter. Once compiled, the programs run from two to twenty times faster, depending on the amount of numeric calculations going on. Commercial software developers are more likely to compile their BASIC programs, since performance is better, and the compiled programs can't be listed out, like interpreted BASIC programs can, thus guarding your proprietary programming techniques.

BASIC Pros and Cons

You don't find many professional programmers who are advocates of BASIC for serious programming. There are many reasons for this, one of which is the "I know a language you don't know" syndrome.

For writing long programs, like a data-base management program or accounting program, Applesoft

BASIC might not be the best friend a programmer can have. Perhaps the biggest objection to BASIC is that it is an unstructured language. This means that you can set up your program in virtually any format you please, provided you follow some simple conventions, such as numbering each program line. While this may be great for short programs, in which you can change the direction of the program while you're in the middle of writing the program at the keyboard, in longer programs you will find yourself in trouble. Without realizing it, you will have likely created a tangled monster of statements that jump all over the place. Trying to trace the operation of the program in your head in search of a "bug," or programming error, can be frustrating even for the one who wrote the program.

In defense of BASIC, it is one of the easiest languages to learn. Commands are largely derived from common English words. And in the interpreted version, you have the ability to learn quickly by experimenting with changes in the program and testing them out immediately without compiling each time.

One thing to remember before committing to BASIC, however, is that once you become comfortable with the unstructured ways of BASIC, it won't be as easy to learn a structured language as it would be if you were raised in another structured language like Pascal.

LOGO

Programming in Logo is fun. Not many languages can claim that, but this highly graphics-oriented language draws even the most passive programming passerby into its simplicity.

The language, a dialect of LISP (the most common language used for artificial intelligence programming), was originally designed at MIT as an educational aid that combined learning of computers and geometry. Commands are largely in English, and, since Logo is an interpreted language, the results of your commands can be viewed instantaneously. At the same time, the language is fully featured enough to stand on its own in productivity-oriented applications. Its strength, however, is as a computer learning tool for the young who have no inhibitions about what a computer is.

What Logo Looks Like

A Logo program listing in no way resembles that of a BASIC program. For one, there are no line numbers. Secondly, you can more easily build a program by writing small modules as you think of them. Perhaps that's the attraction of Logo for learning purposes. As you create a program, you are likely to think up ways to improve or expand it—something Logo lets you do freely. Your program can evolve with your experience with the language.

One popular element of the Logo language, as exemplified in the Apple Logo *//* program, is a feature called turtle graphics. By typing on-screen commands, you control the movement of a make-believe turtle. The turtle drags a pen behind it, so as you instruct the turtle to move forward a certain distance, it leaves a straight line as a trail. If you tell it to turn to the left by 90 degrees (a right angle), it obliges. Forward again, and it has drawn a right angle. Repeating this procedure a few times causes the turtle to draw a square on the screen. With Apple Logo *//*, the ink in the pen can change color on command, giving the programmer the

power to create a screenful of colorful shapes and designs.

Just as BASIC statements in a program run only with the Run command, you can place Logo statements in *procedures* which are executed only when called. The procedure for drawing a square on the screen looks like this:

```
TO SQUARE
REPEAT 4 [FORWARD 30 RIGHT 90]
END
```

In other words, you tell the computer that the procedure, called SQUARE, is performed by repeating the pair of commands, Forward 30 (a relative distance measure on the screen) and Right 90 (degrees), four times. You could use this procedure within another procedure by simply issuing the command SQUARE. In a sense, you help build a library of commands which other procedures can draw from.

Apple Logo *II* also features many commands that allow you to create virtually any kind of program, even if it is not graphics-oriented. Program statements let you print questions on the screen and work with numbers just like any programming language. And because it is so easy to make changes with Logo and see them right away, the language is a growing favorite in schools. Despite its ease of learning, Logo teaches good structured programming habits that can easily carry over to more professional languages like Pascal.

PASCAL

A great many Apple // hobbyists are devotees of Pascal. To them, there is no other language worth knowing. Since most versions of Pascal are compiled versions, the language has the reputation of running rings around BASIC in performance. It is also worth noting that Pascal is an increasingly favorite language with high school and college computer-programming instructors.

Fortunately for the novice programmer, a wealth of Apple Pascal tutorials can be found in book and computer stores. Also, as a rule, you will find other languages, including BASIC, easier to learn if you start out with Pascal.

FORTH

Probably the most vociferous group in support of its own language consists of followers of Forth. The language's first application in 1969 was to help astronomers use computers to control giant telescopes. Since then, Forth has evolved into one of the fastest-operating, most compact (in terms of memory used up by the finished, compiled program) languages on microcomputers. Many commercial programs are written in Forth largely because they can be easily adapted to run on many computer models. Forth is a structured language, like Pascal, but offers the programmer more leeway to improvise the makeup of the program while in the development stage.

CONCLUSIONS

I haven't yet exhausted the programming languages available for the Apple *II*c. But the ones described here are the ones you'll hear most about in your search for the "ideal" programming language for you.

One guideline to follow is that if you plan to program only for yourself, with no intention of getting into the software business, then Applesoft BASIC (and perhaps a BASIC Compiler) will probably be more than adequate for your needs. If learning to program is a family project, then Logo is definitely the language to start with. The kids will be encouraged by the instant graphics response to commands, and the parents will be able to share in the experience because of the language's simplicity. If you think you'll want to learn a language well enough to possibly publish your results, then look more toward the structured languages like Pascal and Forth.

Making it big as a programmer is not a simple task. Be prepared to spend a great deal of time learning the language. Just like a spoken language, the more you use it and learn from your mistakes, the sooner you will begin to feel comfortable with it. No matter how much you learn about a language, you will always find someone who knows more about it than you, or who can perform some programming trick on a piece of software that you can't figure out. Don't let that discourage you. No one knows it all. The little tricks come with time and with contact with fellow programmers in users' groups and on-line bulletin boards.

APPENDICES

APPENDIX A
Apple *II*c Technical Specifications

Keyboard:
63 keys; 2-key rollover; automatic repeat; 128-character ASCII encoded

Front-panel Switches:
40/80 column video display (can be overridden by software); Sholes/Dvorak keyboard

Front-panel Lights:
Power on; disk in use

Internal Disk Drive:
Single-sided, $5^1/_4$-inch diameter, 35 tracks, 16 sectors per track

Rear-panel Connectors:
Power—7-pin DIN-type
Serial Port #1 (Printer)—5-pin D-type
Second Disk Drive—19-pin D-type
Composite Video Monitor—RCA-type phono jack
Video Expansion (color TV et al.)—15-pin D-type
Serial Port #2 (Modem)—5-pin D-type
Hand controls/mouse—9-pin D-type miniature

Power Supply:

Remote floor-mounted AC-to-DC transformer, 105–129 volts AC input, providing 15 volts DC to the internal voltage convertor; power consumption—25 watts continuous

Operating Environment Ranges:

10 to 40 degrees C (50 to 104 degrees F); 20 percent to 95 percent relative humidity

Microprocessor:

65C02, CMOS version of 6502, running at 1.023 MHz, performing up to 500,000 8-bit operations per second

RAM:

128K on-board, bank-switched; not expandable

ROM:

16K; includes Applesoft BASIC and Monitor

Custom LSI Chips:

Memory Management Unit (MMU)—controls memory addressing

Input/Output Unit (IOU)—controls internal input/output functions

Timing Generator (TMG)—generates system and I/O timing signals derived from a 14.31818 MHz oscillator

General Logic Unit (GLU)—other logic functions

Integrated "Woz" Machine (IWM)—single-chip equivalent of the Apple Disk *//* controller card

Video Display Modes:

40-column text (40 columns by 24 lines)

80-column text (80 columns by 24 lines)

Low-resolution color graphics (40 horiz. by 48 vert., 16 colors)

High-resolution color graphics (280 horiz. by 192 vert., 6 colors)
Double-high-resolution color graphics (560 horiz. by 192 vert., 16 colors)

Text Character Sets:
Normal—white dots on black screen
Inverse—black dots on white screen
Flashing—alternating normal/inverse
Mouse Text—32 graphics characters in ROM (occupying ROM space of alternate Inverse character set in earlier Apple *//* models)

Serial Port #1 Default Settings:
9600 baud; 8 data bits, no parity bits, 2 stop bits; 80-column line width; echo off; linefeed with carriage return

Serial Port #2 Default Settings:
300 baud; 7 data bits, no parity bits, 1 stop bit; echo off; no linefeed with carriage return

Serial Ports Setting Ranges:
Line width—1 to 255
Baud rate—50, 75, 110, 135, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200
Data/Stop bits—8/1, 7/1, 6/1, 5/1, 8/2, 7/2, 6/2, 5/2
Echo—on/off
Linefeed (LF) on carriage return (CR)—on/off
Parity—none, odd, even, mark, space

APPENDIX B
Apple *II*c Rear Panel Connections

Since the only access peripheral designers have to the Apple *II*c system is by way of the rear panel connections, the signals carried at the pins of those connectors are important. The same is true if you wish to configure a connecting cable between a non-Apple peripheral and one of the connectors. The information presented below is derived from the Apple *II*c *Technical Reference Manual*.

VIDEO EXPANSION CONNECTOR SIGNALS

Pin Number	Name	Description
1	TEXT	Video text signal from GLU
2	14M	Master timing signal from 14 MHz oscillator
3	SYNC′	Display Synchronization signal
4	SEGB	Display vertical counter bit
5	IVSOUND	Sound signal
6	LDPS′	Video shift-register load enable
7	WNDW′	Active area display blanking
8	+12V	Regulated +12 volts
9	PRAS′	RAM row-address strobe
10	GR	Graphics mode enable
11	SEROUT′	Serialized character generator output
12	NTSC	Composite NTSC video signal
13	GND	Ground
14	VIDD7	Half-dot shift
15	CREF	Color reference signal

DISK DRIVE CONNECTOR SIGNALS

Pin Number	Name	Description
1,2,3,4	GND	Ground
6,16	+5V	+5 volts
7,8	+12V	+12 volts
9	EXTINT′	External interrupt
10	WRPROT	Write protect input
11–14	PHØ-4	Motor phase output
15	WRREQ′	Write request
17	DR1′	Drive 1 select
18	RDDATA	Read data input
19	WRDATA	Write data output

SERIAL PORT CONNECTORS SIGNALS

Pin Number	Name	Description
1	DTR	Data terminal ready
2	TD	Transmit data
3	GND	Ground
4	RD	Read Data
5	DSR	Data Set Ready

MOUSE CONNECTOR SIGNALS

Pin Number	Name	Description
1	MOUSEID′	Mouse identifier (disables hand controller chip when active)
2	+5V	+5 volts
3	GND	System ground
4	X1	X direction indicator
5	XØ	X movement interrupt
6		Mouse button
7	MSW′	Mouse button
8	Y1	Y direction indicator
9	YØ	Y movement indicator

HAND CONTROL CONNECTOR SIGNALS

Pin Number	Name	Description
7,1,6	SW1,SWØ	Paddle buttons
2	+5V	+5 volts
3	GND	System Ground
5,8,4,9	PDLØ-PDL3	Hand control inputs

APPENDIX C
Differences Between Apple *//c* and Earlier Apple *//s*

If you are familiar with earlier Apple II computers, you can see below the evolution of the Apple *//* family from the original II to the II Plus, the *//e*, and now the new *//c*. For each model after the original Apple II, I have listed the features that were added or subtracted. Consult the various sections of this book for more details about each of the features as implemented on the *//c*.

I. To turn an Apple II into an Apple II Plus:

ADD	SUBTRACT
Autostart ROM	Old Monitor ROM
Applesoft firmware	Integer BASIC firmware
48K total on-board RAM	

II. To turn an Apple II Plus into an Apple //e:

ADD	SUBTRACT
Language Card (incl. 16K RAM)	Slot Ø
80-column firmware	
Built-in diagnostics	
Full ASCII keyboard	
Internal power-on light	
FCC approval	
Rear panel holes for connectors	
9-pin rear panel game connector	
Auxiliary slot (for 80-column/64K RAM card)	

III. To turn an Apple *II*e into an Apple *II*c:

ADD	SUBTRACT
80-column/64K RAM card	Removable cover
40/80 column switch	Slots 1 to 7
Keyboard switch	Auxiliary slot
Disk light	Internal power-on light
Disk controller port	Cassette I/O connectors
Disk drive	Internal game I/O connector
Mouse port	RF modulator connector
Serial printer port	Auxiliary video pin
Serial communication port	Diagnostic firmware
Built-in port firmware	Monitor cassette support
Video expansion connector	DOS 3.2 support
Mouse Text characters	

The most fundamental hardware differences between the Apple *II*e and *II*c are related to peripheral ports, which are built into the *II*c. Peripherals like the external disk drive are addressable by specifying certain slot numbers in programs that ask for that information. The built-in disk drive is equivalent to Slot 6, Drive 1; the external disk is equivalent to Slot 6, Drive 2.

The 65C02 microprocessor offers 27 new programming instructions. If any software uses those instructions, it will not run on earlier Apple *II*s, since the 6502 won't know how to respond. Apparently, you can plug a 65C02 into the microprocessor socket of an Apple *II*e (but not earlier models) to gain more compatibility with *II*c-specific software.

The *II*c continues the trend toward fewer chips on the main board. This is accomplished by integrating more functions onto fewer custom large-scale integrated chips. While just about every chip on the Apple *II*e motherboard was inserted into a socket, many of the *II*c chips, including all RAMs, are soldered to the board. Apple engineers believe this will increase product reliability still further.

One subtraction from the *II*e, the diagnostics routine, is superfluous in the *II*c. Apple *II*e dealers used the diagnostics program to make sure the computer operated properly once they fiddled with inserting expansion cards into the slots. With the *II*c, however, the dealer doesn't need to unpack and inspect a customer's machine before it leaves the store.

If you own an earlier Apple *II* model and want to advance to the *II*c yet still hang on to your software, you will have mixed results. As with the differences between the II Plus and *II*e, software programs that use copy protection schemes that rely on ROM bugs may not work on the *II*c if that area of the ROM has been used for other things. You may also have difficulty taking your disks to a dealer and trying them out on the *II*c if they are formatted in DOS 3.2. The *II*c

recognizes only DOS 3.3, Pro-DOS, and Pascal disks. A conversion utility program is provided with the computer (see Appendix D), so perhaps a patient dealer would let you convert copies of your important disks to try out.

APPENDIX D
The System Utilities Disk

One of the disks that comes packaged in the Apple *II*c box is labeled System Utilities. One of the three manuals also packed with the computer is dedicated to this disk. A system utility is a program that helps you manage your disks and perform a few other functions. Utilities are the programs that let you prepare blank disks for storage (formatting) and make backup copies of disks.They operate in conjunction with a special program that loads into your computer every time you use a disk program: the Disk Operating System, or DOS.

You can start up BASIC by turning on the *II*c without a disk in the drive and then pressing the Control and Reset buttons simultaneously. But since the DOS has not been loaded into the computer—a procedure that occurs automatically when you start the computer with a disk in the drive—the computer doesn't have the instructions needed to let you read from or write to disks.

When you turn on the *II*c without a disk in the drive (and without pressing the buttons to get in into

BASIC), you get an advisory on the screen to check your disk drive. That's because the computer's ROM instructions tell it to find a disk and load DOS.

The Apple *II*c can function under three different Disk Operating Systems: DOS 3.3, Pro-DOS, and one called Pascal (the same name as the programming language). Each operating system has different functions, commands, and characteristics, which could cause a newcomer to have a hard time figuring out how to perform some needed disk management maneuvers, like copying a file from disk to disk. Moreover, if you have applications programs written in different operating systems and need storage disks formatted, you have to make sure you format each storage disk in the proper operating system, or it won't work with the program. That's where the new System Utilities disk comes in handy.

Built into the System Utilities disk are most of the disk management features from all three operating systems. You select the operation you want to perform, and the program makes all the determinations about which operating system you need. The main menu of the disk provides these functions:

Copy Files
Delete Files
Rename Files
Lock/Unlock Files
Duplicate a Disk
Format a Disk
Indentify and Catalog a Disk

Except for formatting a blank disk, you don't have to worry about which operating system you're using. For example, if you want to copy a file from one disk to another, simply select the Copy Files option from the main menu and answer questions about which disk drive the source and target disks are located. The program reads the disk you have in the source drive and lists the names of all the files on the disk. Next, you select the file(s) to copy by placing a check mark next to each file to be copied (you use the arrow keys to move the pointer and place check marks). After that, the program makes the copy according to the rules of the source disk's operating system.

Formatting a disk brings the operating system differences more into view, but not so much as to make things confusing. Selecting the Format a Disk option from the main menu presents a second menu that asks in which DOS you want the blank disk formatted. If you're not sure in which DOS your applications program is written, you can select one of the menu responses that lets you place the program disk in the drive and have the computer figure out which operating system you need.

Another feature of the System Utilities disk is an operation that changes the formats of disks. This is most useful for owners of earlier Apple *II* models who have disks and files in the older DOS 3.2 version. The Apple *II*c does not recognize DOS 3.2 files or disks directly, so this utility is essential to making the disks work on the *II*c.

One last important function of the System Utilities disk is the reconfiguration of the serial ports on the *II*c.

You may have noticed in Appendix A that the two serial ports have their parameters set so that port #1 can be used with a printer (the Apple Imagewriter, specifically) and port #2 with a 300-baud modem. If you use a 1200-baud modem or want to attach serial devices with parameters other than the default settings, you use the System Utilities disk to reconfigure the ports as needed. A series of menus lead you through all the elements that the computer needs to know about the device.

Once the parameters for a device are set, they are placed in a library and the name of the device then appears on the on-screen menu (available when the System Utilities disk is in the drive), from which you can readily select your device. Some applications programs have their own serial port configuration routines built in, so you won't always have to load the System Utilities disk each time you want to run a program using a nonstandard peripheral.

The implementation of the System Utilities disk is really quite a breakthrough in simplifying the otherwise technical nature of personal computers. It is a big step closer to the operating system invisibility (often called "transparency") designed into the Macintosh. It's nice, for a change, to have the computer do the drudgery for us.

APPENDIX E

For More Information

Aardvark
783 N. Water St.
Milwaukee, WI 53202

Apple Computer
20525 Mariani Ave.
Cupertino, CA 95014

Broderbund Software
17 Paul Drive
San Rafael, CA 94903

CBS Software
1 Fawcett Place
Greenwich, CT 06836

CompuServe
5000 Arlington Centre Blvd
Columbus, OH 43220

Datasoft
19808 Nordhoff Place
Chatsworth, CA 91311

Dialog Information
Services, Inc.
3460 Hillview Avenue
Palo Alto, CA 94304

Dow Jones Information
Services
Box 300
Princeton, NJ 08540

Electronic Arts
2755 Campus Drive
San Mateo, CA 94403

Infocom, Inc.
P.O. Box 855
Garden City, NY 11530

Living Videotext Inc.
1000 Elwell Court
Palo Alto, CA 94303

Microsoft Corporation
10700 Northrup Way
Bellevue, WA 98004

Odesta
3186 Doolittle Drive
Northbrook, IL 60062

Scarborough Systems
225 N. Broadway
Tarrytown, NY 10591

Scholastic, Inc.
730 Broadway
New York, NY 10003

Simon & Schuster/
Electronic Publishing
1230 Avenue of the
Americas
New York, NY 10020

Software Publishing
Corporation
1901 Landings Dr.
Mountain View, CA 94043

Spinnaker Software Co.
215 First Street
Cambridge, MA 02142

The Learning Co.
545 Middlefield Rd.
Menlo Park, CA 94025

The Source Telecomputing
Corp.
1616 Anderson Rd.
MacLean, VA 22102

Xerox Education
Publications
245 Long Hill Rd.
Middletown, CT 06457

LEAD. IIC. FOR IMAGEWRITER.

1	DTR.	6	DSR
2	TD	7	SG
3	GND.	20	[illegible]
4	RD	3	[illegible]
5.	D[illegible].	2	SD.

What new computer weighs 7½ pounds, is fully transportable, contains a whopping 128K memory, runs thousands of different software packages, creates dazzling graphics and costs just a fraction of whatever you imagined such a machine should cost?

ANNOUNCING THE REVOLUTIONARY
Apple IIc

Personal computing enters a new era with the Apple IIc, the most dramatic addition to the Apple II family and the ultimate personal tool for every man, woman and child of the Eighties. In this exclusive insider's look Danny Goodman reveals everything that a IIc can do for you. With a clear, comprehensive look at the machine and its new and many unique features and add-on devices, its stunning software (including MousePaint by game designer Bill Budge) Goodman shows the tremendous variety of practical, educational, entertaining and artistic things you can do. But most important, you'll discover what the excitement is all about, and why–unlike any personal computer before it–the Apple IIc is the perfect one for you.

53188
0 76714 00395
ISBN 0-671-53188-3